

---

# 練習

## Restful API

---

鄭安翔

ansel\_cheng@hotmail.com

# Lab

## ■ ContactInfo 網路應用專案

- 使用 Rest Web Service API 新增/修改/刪除客戶資料
- 不包含Filter機制 (與第九章練習完成結果相同)
- 修改網路伺服器為 TomEE

# Lab

## ■ 修改 Customer.java

- 新增無參數建構子
- 刪除靜態測試資料及靜態方法
  - 搬移至CustomerResource.java
- 新增下列 **setter** 方法
  - `public void setOfficeAddress(Address address)`
  - `public void setBillingAddress(Address address)`
  - `public void setDeliveryAddress(Address address)`
- 刪除取得所有地址的方法
  - `public Address[] getAddresses() {`

# Lab

- 修改 `customerList.jsp`
  - EL 不使用 `Addresses[0]` 顯示客戶聯絡資料
    - 改用 `OfficeAddress`, `BillingAddress`, `DeliveryAddress` 屬性
- 修改 `customerView.jsp`
  - EL 不使用 `Addresses[2]` 顯示客戶聯絡資料
    - 改用 `customer.deliveryAddress` 屬性

# Lab

- Restful Web Service Resource
  - 新增 Restful Web Service Resource
    - File Type
      - Restful Web Service From Patterns
    - Pattern
      - Design Pattern: Single Root Resource
    - Resource Class
      - Package: service
      - Path: customer
      - ClassName: customerResource
      - MIME Type: application/json

# Lab

- ❑ 檢視 ApplicationConfig.java

- ApplicationPath: webresources
- addRestResourceClasses() 包含
  - ❑ service.CustomerResource.class

- ❑ CustomerResource.java

- 將Customer 的靜態測試資料及靜態方法搬移過來
  - ❑ HashMap<Integer, Customer> customers
  - ❑ static {...}
  - ❑ public static Customer getCustomer(int id)
  - ❑ public static Collection<Customer> getCustomers()

# Lab

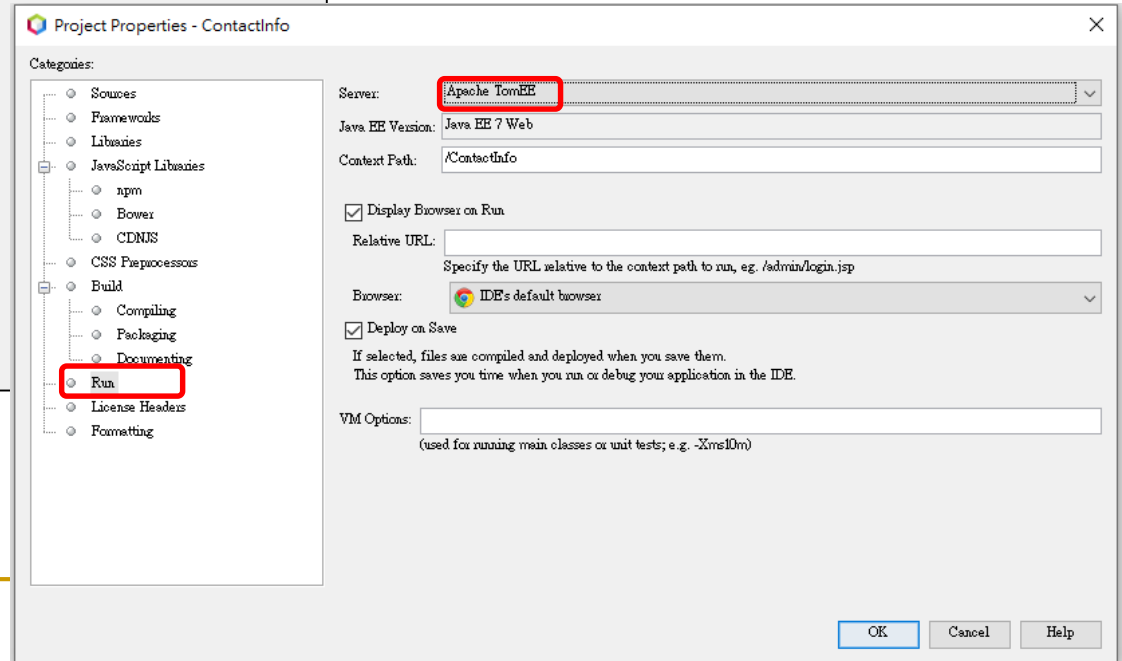
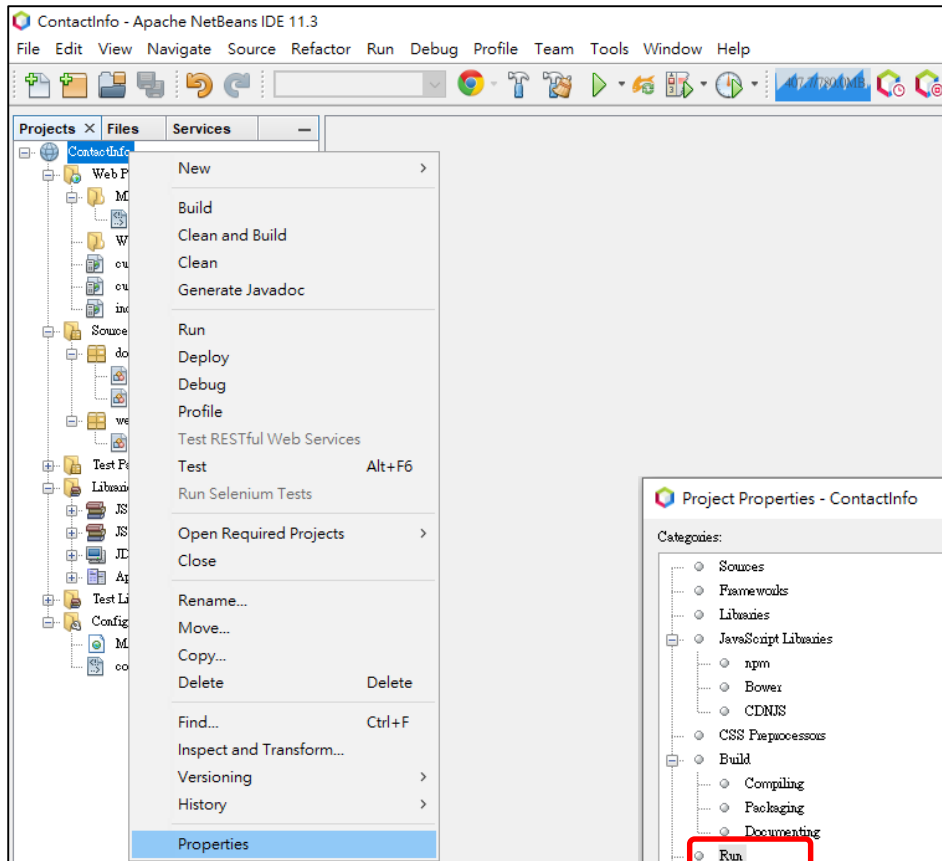
- Response addCustomer(@PathParam("id")Integer id, Customer customer) 方法
  - @POST / @Path("{id}") / @Consumes(MediaType.APPLICATION\_JSON)
  - 客戶編號已存在 customers 中, 產生錯誤回應, 代碼400
  - 客戶編號不存在, customers 中新增客戶 put(id, customer), 產生成功回應, 傳回 customers
- Response updateUser(@PathParam("id")Integer id, Customer customer) 方法
  - @PUT / @Path("{id}") / @Consumes(MediaType.APPLICATION\_JSON)
  - 客戶編號不存在 customers 集合中, 產生錯誤回應, 代碼400
  - 客戶編號已存在, customers 中取代客戶 put(id, customer), 產生成功回應, 傳回 customers

# Lab

- Response deleteUser(@PathParam("id")Integer id) 方法
  - @DELETE / @Path("{id}") / @Consumes(MediaType.APPLICATION\_JSON)
  - 客戶編號不存在 customers 集合中, 產生錯誤回應, 代碼400
  - 客戶編號不存在, customers 中 remove(id) 移除客戶), 產生成功回應, 傳回 customers
- 修改 web.Controller.java
  - CustomerResource.getCustomers() 取得所有客戶
  - CustomerResource.getCustomer(custid) 取得指定客戶
- 測試、執行



# ContactInfo 專案



# Customer.java

```
Customer.java
Source History
1 package domain;
2
3 public class Customer {
4     private String name;
5     public static final int HEAD_OFFICE = 0;
6     public static final int BILLING = 1;
7     public static final int DELIVERY = 2;
8     private Address [] addresses = new Address[3];
9
10    public Customer() {
11    }
12
13    public Customer(String name, Address office, Address billing, Address delivery) {
14        this.name = name;
15        addresses[HEAD_OFFICE] = office;
16        addresses[BILLING] = billing;
17        addresses[DELIVERY] = delivery;
18    }
19
20    // 測試資料
21    private static final HashMap<Integer, Customer> customers = new HashMap<Integer, Customer>();
22    static {
23        customers.put(1, new Customer("蔡英文",
24            new Address("凱達格蘭大道1號", "總統府廣場", "台北", "台灣", "100"),
25            new Address("重慶南路一段122號", "總統府", "台北", "台灣", "100"),
26            new Address("重慶南路二段100號", "總統官邸", "台北", "台灣", "100")));
27        customers.put(2, new Customer("柯文哲",
28            new Address("市府路1號", "台北市政府", "台北", "台灣", "110"),
29            null,
30            new Address("徐州路46號", "台北市長官邸", "台北", "台灣", "100")));
31    }
32    public static Customer getCustomer(int id) {
33        return customers.get(id);
34    }
35    public static Collection<Customer> getCustomers() {
36        return customers.values();
37    }
38 }
```

```
39 public String getName() {
40     return name;
41 }
42
43 public void setName(String name){
44     this.name = name;
45 }
46
47 public Address getOfficeAddress() {
48     return addresses[HEAD_OFFICE];
49 }
50
51 public void setOfficeAddress(Address address){
52     addresses[HEAD_OFFICE] = address;
53 }
54
55 public Address getBillingAddress() {
56     return addresses[BILLING];
57 }
58
59 public void setBillingAddress(Address address){
60     addresses[BILLING] = address;
61 }
62
63 public Address getDeliveryAddress() {
64     return addresses[DELIVERY];
65 }
66
67 public void setDeliveryAddress(Address address){
68     addresses[DELIVERY] = address;
69 }
70
71 public Address[] getAddresses() {
72     return addresses;
73 }
74
75 }
```

customerView.jsp

```

10 <body>
11   <h1>客戶資訊</h1>
12   <c:choose>
13     <c:when test = "${not empty customer}">
14       <p>姓名: ${customer.name}</p>
15       <cif test = "${not empty customer.officeAddress}">
16         <p>辦公室地址:<br>
17           ${customer.officeAddress.address1},<br>
18           ${customer.officeAddress.address2},<br>
19           ${customer.officeAddress.city},<br>
20           ${customer.officeAddress.country},<br>
21           ${customer.officeAddress.postcode}<br>
22         </p>
23       </cif>
24       <cif test = "${customer.billingAddress ne null}">
25         <p>發票地址:<br>
26           ${customer["billingAddress"].address1},<br>
27           ${customer["billingAddress"].address2},<br>
28           ${customer["billingAddress"].city},<br>
29           ${customer["billingAddress"].country},<br>
30           ${customer["billingAddress"].postcode}<br>
31         </p>
32       </cif>
33       <cif test = "${not empty customer.deliveryAddress}">
34         <p>送貨地址:<br>
35           ${customer.deliveryAddress.address1},<br>
36           ${customer.deliveryAddress.address2},<br>
37           ${customer.deliveryAddress.city},<br>
38           ${customer.deliveryAddress.country},<br>
39           ${customer.deliveryAddress.postcode}<br>
40         </p>
41       </cif>
42     </c:when>
43     <c:otherwise>
44       <h3>查詢的客戶不存在</h3>
45     </c:otherwise>
46   </c:choose>
47 </body>

```

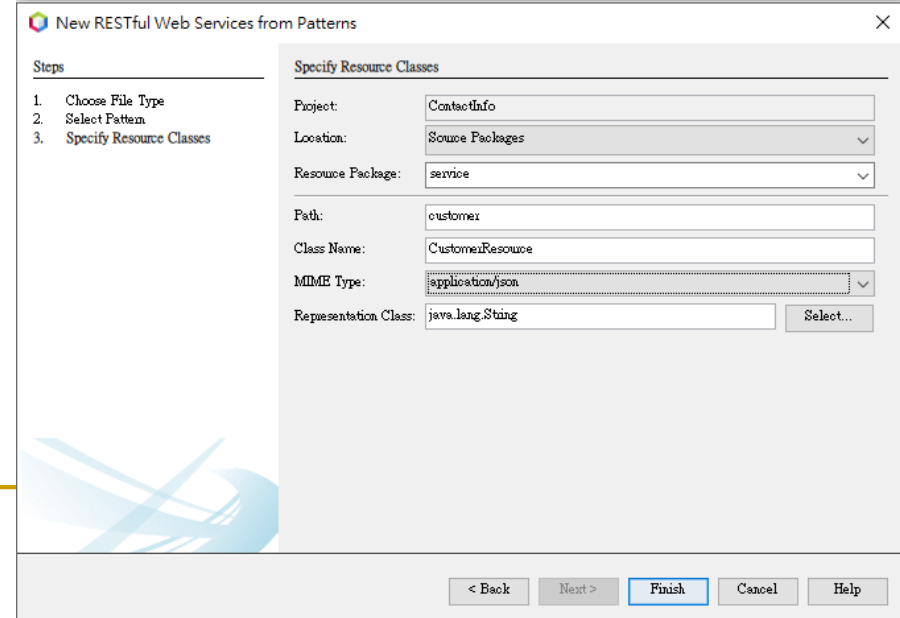
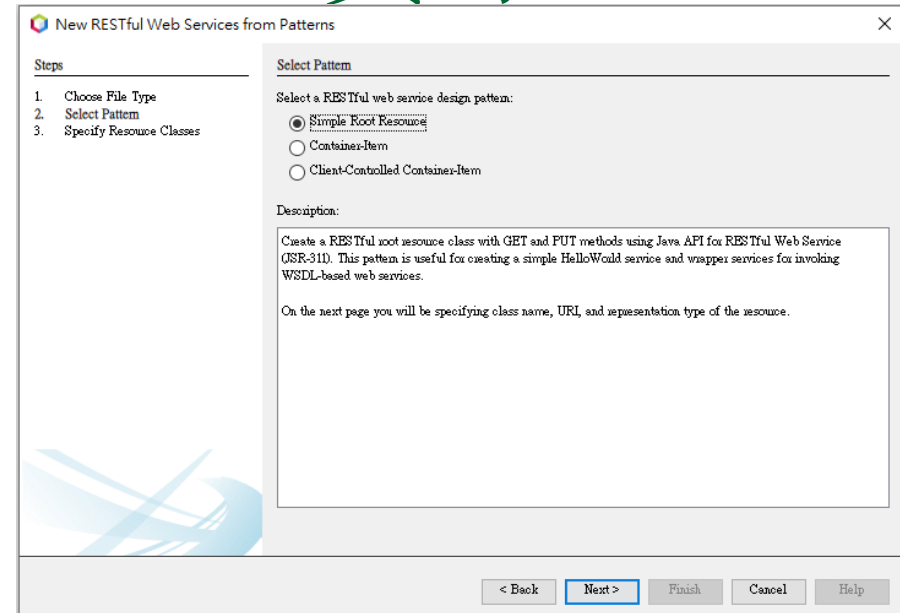
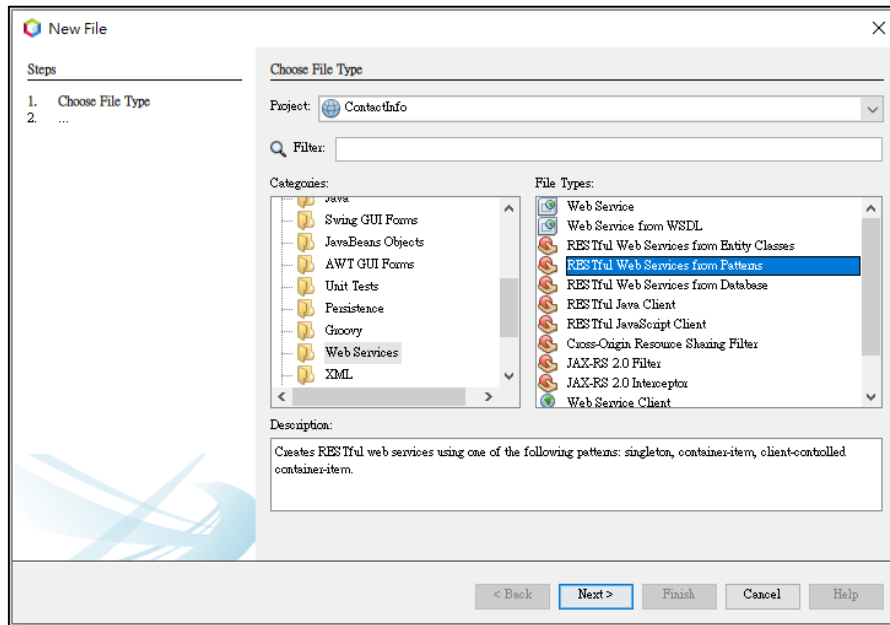
customerList.jsp

```

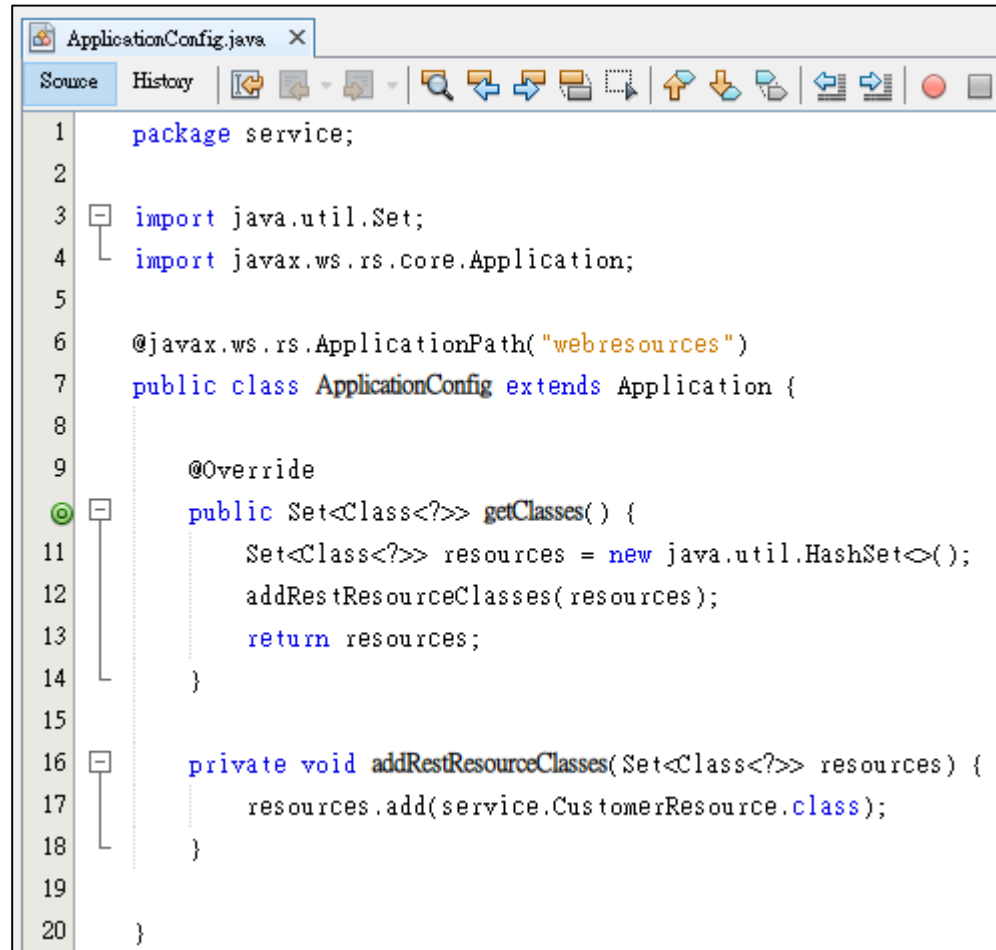
9 <body>
10   <h1>客戶列表</h1>
11   <c:choose>
12     <c:when test = "${not empty customers}">
13       <c:forEach var="customer" items="${customers}">
14         <p>姓名: ${customer.name}</p>
15         <cif test = "${not empty customer.officeAddress}">
16         <p>辦公室地址:
17           ${customer.officeAddress.address1},
18           ${customer.officeAddress.address2},
19           ${customer.officeAddress.city},
20           ${customer.officeAddress.country},
21           ${customer.officeAddress.postcode}
22         </p>
23         </cif>
24         <cif test = "${customer.billingAddress ne null}">
25         <p>發票地址:
26           ${customer["billingAddress"].address1},
27           ${customer["billingAddress"].address2},
28           ${customer["billingAddress"].city},
29           ${customer["billingAddress"].country},
30           ${customer["billingAddress"].postcode}
31         </p>
32         </cif>
33         <cif test = "${not empty customer.deliveryAddress}">
34         <p>送貨地址:
35           ${customer.deliveryAddress.address1},
36           ${customer.deliveryAddress.address2},
37           ${customer.deliveryAddress.city},
38           ${customer.deliveryAddress.country},
39           ${customer.deliveryAddress.postcode}
40         </p>
41         </cif>
42       </c:forEach>
43     </c:when>
44     <c:otherwise>
45       <h3>無客戶存在</h3>
46     </c:otherwise>
47   </c:choose>
48 </body>

```

# 新增 Rest Web Service 資源 Customer

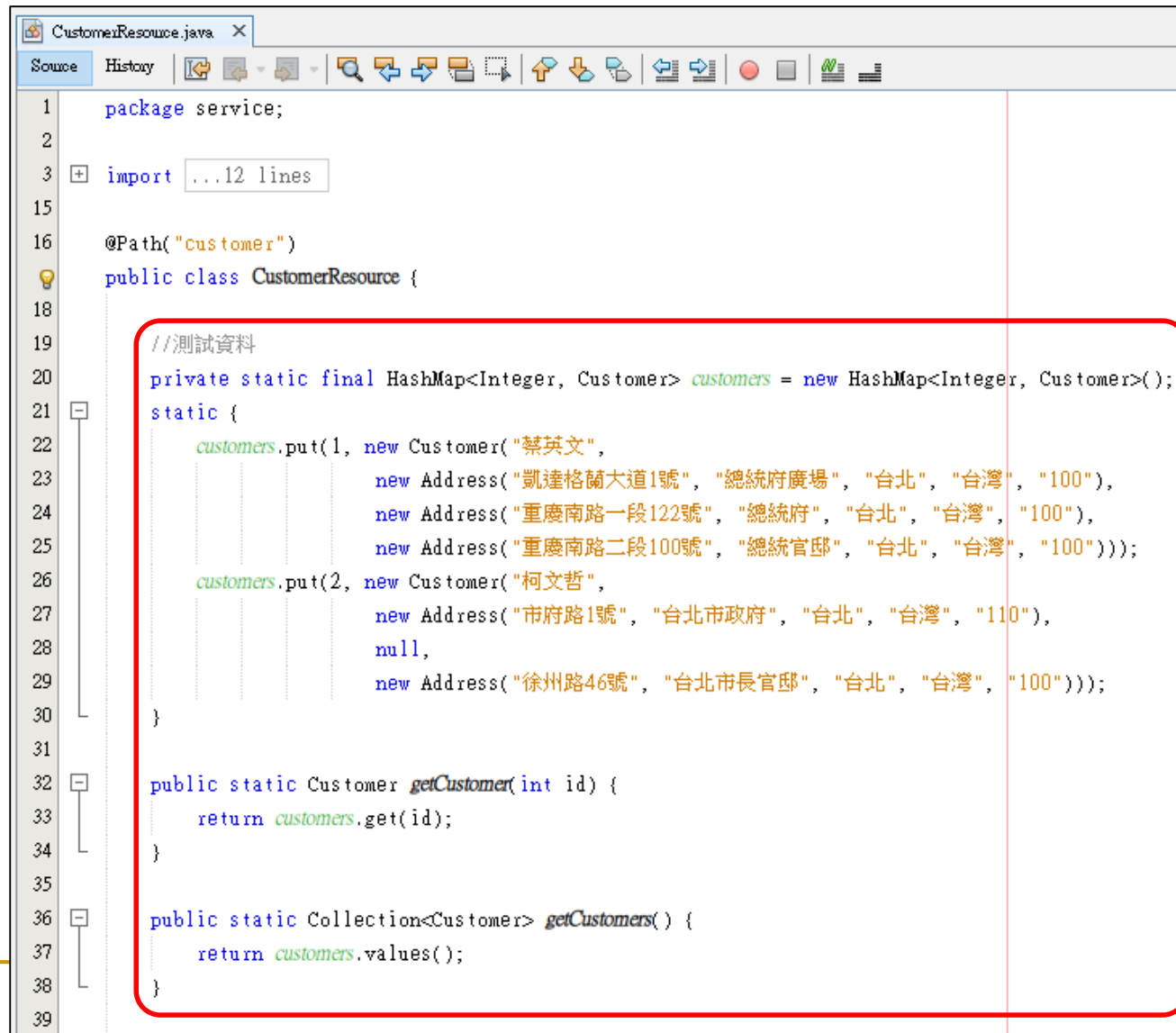


# ApplicationConfig.java



```
1  package service;
2
3  import java.util.Set;
4  import javax.ws.rs.core.Application;
5
6  @javax.ws.rs.ApplicationPath("webresources")
7  public class ApplicationConfig extends Application {
8
9      @Override
10     public Set<Class<?>> getClasses() {
11         Set<Class<?>> resources = new java.util.HashSet<>();
12         addRestResourceClasses(resources);
13         return resources;
14     }
15
16     private void addRestResourceClasses(Set<Class<?>> resources) {
17         resources.add(service.CustomerResource.class);
18     }
19
20 }
```

# CustomerResource.java



```
1 package service;
2
3 import ...12 lines
4
5
6
7
8
9
10
11
12
13
14
15
16 @Path("customer")
17 public class CustomerResource {
18
19     //測試資料
20     private static final HashMap<Integer, Customer> customers = new HashMap<Integer, Customer>();
21     static {
22         customers.put(1, new Customer("禁英文",
23             new Address("凱達格蘭大道1號", "總統府廣場", "台北", "台灣", "100"),
24             new Address("重慶南路一段122號", "總統府", "台北", "台灣", "100"),
25             new Address("重慶南路二段100號", "總統官邸", "台北", "台灣", "100")));
26         customers.put(2, new Customer("柯文哲",
27             new Address("市府路1號", "台北市政府", "台北", "台灣", "110"),
28             null,
29             new Address("徐州路46號", "台北市長官邸", "台北", "台灣", "100")));
30     }
31
32     public static Customer getCustomer(int id) {
33         return customers.get(id);
34     }
35
36     public static Collection<Customer> getCustomers() {
37         return customers.values();
38     }
39 }
```

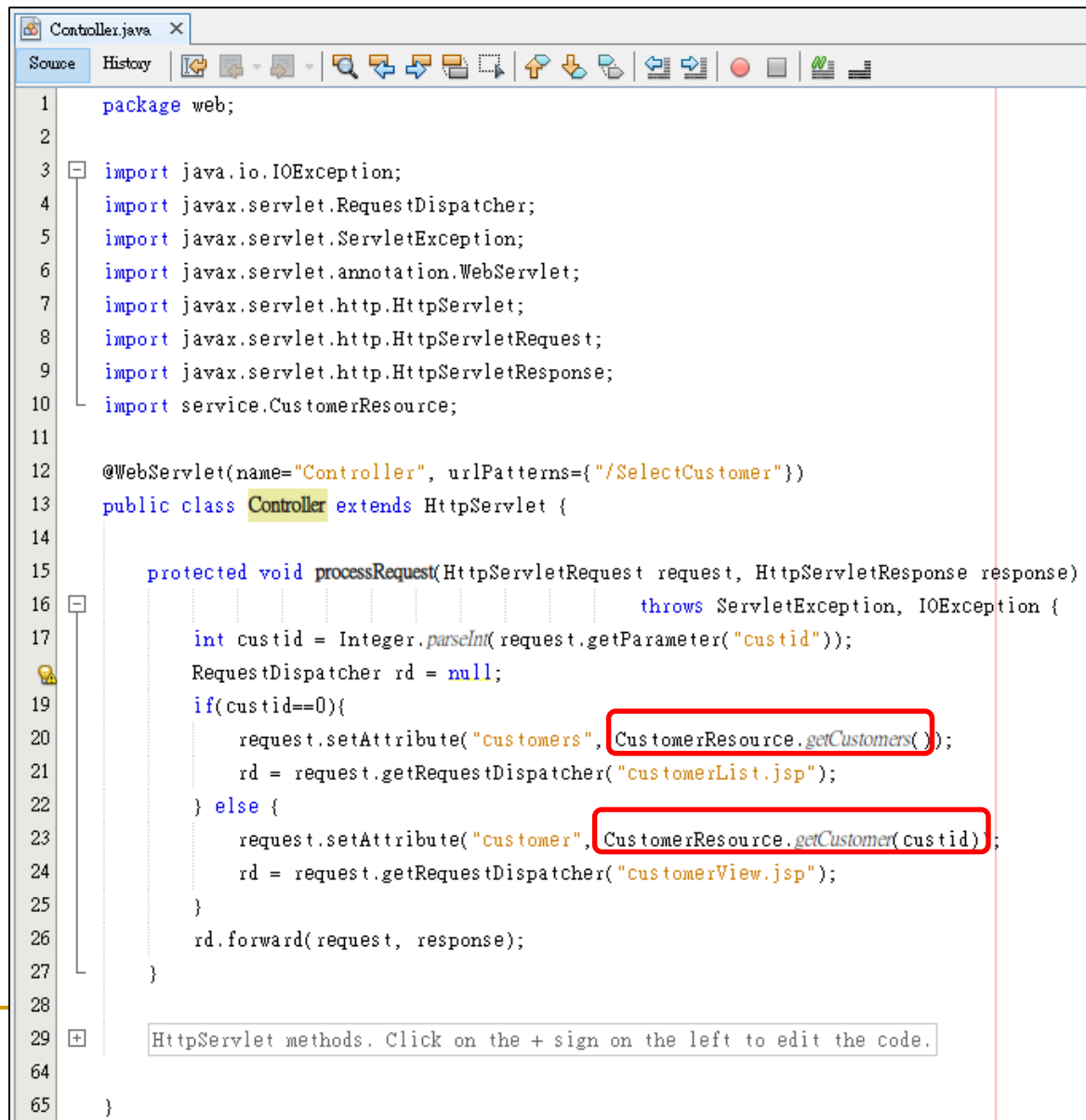
# CustomerResource.java

```
40     @Path("/{id}")
41     @POST
42     @Consumes(MediaType.APPLICATION_JSON)
43     public Response addCustomer(@PathParam("id") Integer id, Customer customer){
44         if(customers.containsKey(id)){
45             return Response.status(400).build();
46         } else {
47             customers.put(id, customer);
48             return Response.ok()
49                 .header("status", "ok")
50                 .encoding("UTF-8")
51                 .entity(customers)
52                 .build();
53         }
54     }
55 }
```

```
56     @Path("/{id}")
57     @PUT
58     @Consumes(MediaType.APPLICATION_JSON)
59     public Response updateUser(@PathParam("id") Integer id, Customer customer){
60         if(customers.containsKey(id)){
61             customers.put(id, customer);
62             return Response.ok()
63                 .header("status", "ok")
64                 .encoding("UTF-8")
65                 .entity(customers)
66                 .build();
67         } else {
68             return Response.status(400).build();
69         }
70     }
71 }
```

```
72     @Path("/{id}")
73     @DELETE
74     @Consumes(MediaType.APPLICATION_JSON)
75     public Response deleteUser(@PathParam("id") Integer id){
76         if(customers.containsKey(id)){
77             customers.remove(id);
78             return Response.ok()
79                 .header("status", "ok")
80                 .encoding("UTF-8")
81                 .entity(customers)
82                 .build();
83         } else {
84             return Response.status(400).build();
85         }
86     }
87 }
```

# web.Controller.java



```
1 package web;
2
3 import java.io.IOException;
4 import javax.servlet.RequestDispatcher;
5 import javax.servlet.ServletException;
6 import javax.servlet.annotation.WebServlet;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10 import service.CustomerResource;
11
12 @WebServlet(name="Controller", urlPatterns={"/SelectCustomer"})
13 public class Controller extends HttpServlet {
14
15     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17         int custid = Integer.parseInt(request.getParameter("custid"));
18         RequestDispatcher rd = null;
19         if(custid==0){
20             request.setAttribute("customers", CustomerResource.getCustomers());
21             rd = request.getRequestDispatcher("customerList.jsp");
22         } else {
23             request.setAttribute("customer", CustomerResource.getCustomer(custid));
24             rd = request.getRequestDispatcher("customerView.jsp");
25         }
26         rd.forward(request, response);
27     }
28
29     HttpServlet methods. Click on the + sign on the left to edit the code.
30
31 }
```



# 測試、執行

Postman interface showing a POST request to `http://localhost:8080/ContactInfo/webresources/customer/3`. The request body is a JSON object:

```
{
  "name": "蘇貞昌",
  "officeAddress": {
    "address1": "忠孝東路1段1號",
    "address2": "行政院",
    "city": "台北市",
    "country": "台灣",
    "postcode": "100"
  },
  "deliveryAddress": {
    "address1": "金華街142號",
    "address2": "行政院長官邸",
    "city": "台北市",
    "country": "台灣",
    "postcode": "100"
  }
}
```

The response is a JSON object:

```
{
  "3": {
    "deliveryAddress": {
      "address1": "金華街142號",
      "address2": "行政院長官邸",
      "city": "台北市",
      "country": "台灣",
      "postcode": "100"
    },
    "name": "蘇貞昌",
    "officeAddress": {
      "address1": "忠孝東路1段1號",
      "address2": "行政院",
      "city": "台北市",
      "country": "台灣",
      "postcode": "100"
    }
  }
}
```

Customer List

姓名: 蔡英文

辦公室地址: 凱達格蘭大道1號, 總統府廣場, 台北, 台灣, 100

發票地址: 重慶南路一段122號, 總統府, 台北, 台灣, 100

送貨地址: 重慶南路二段100號, 總統官邸, 台北, 台灣, 100

姓名: 柯文哲

辦公室地址: 市府路1號, 台北市政府, 台北, 台灣, 110

送貨地址: 徐州路46號, 台北市長官邸, 台北, 台灣, 100

Customer List

姓名: 蔡英文

辦公室地址: 凱達格蘭大道1號, 總統府廣場, 台北, 台灣, 100

發票地址: 重慶南路一段122號, 總統府, 台北, 台灣, 100

送貨地址: 重慶南路二段100號, 總統官邸, 台北, 台灣, 100

姓名: 柯文哲

辦公室地址: 市府路1號, 台北市政府, 台北, 台灣, 110

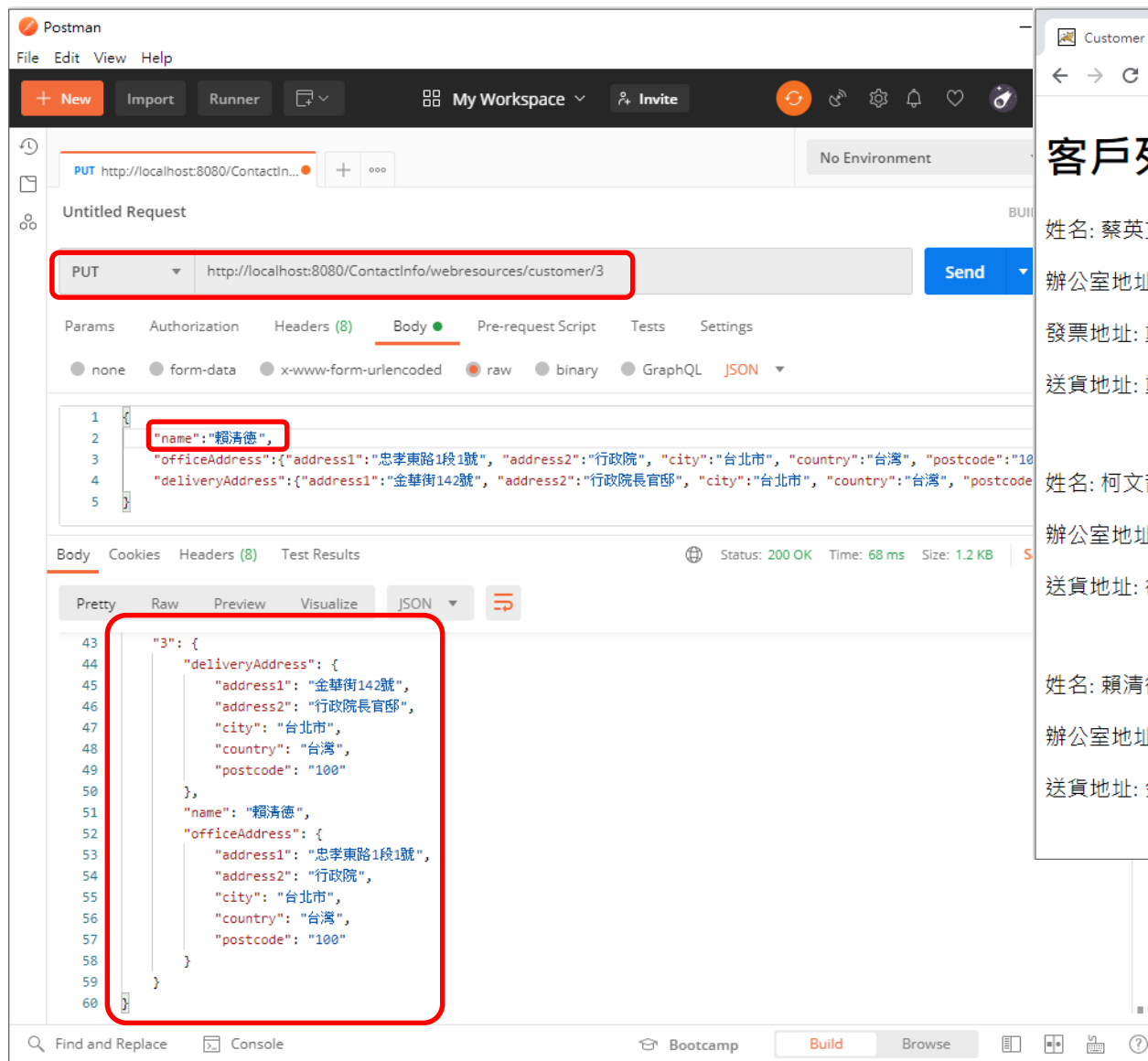
送貨地址: 徐州路46號, 台北市長官邸, 台北, 台灣, 100

姓名: 蘇貞昌

辦公室地址: 忠孝東路1段1號, 行政院, 台北市, 台灣, 100

送貨地址: 金華街142號, 行政院長官邸, 台北市, 台灣, 100

# 測試、執行



## 客戶列表

姓名: 蔡英文

辦公室地址: 凱達格蘭大道1號, 總統府廣場, 台北, 台灣, 100

發票地址: 重慶南路一段122號, 總統府, 台北, 台灣, 100

送貨地址: 重慶南路二段100號, 總統官邸, 台北, 台灣, 100

姓名: 柯文哲

辦公室地址: 市府路1號, 台北市政府, 台北, 台灣, 110

送貨地址: 徐州路46號, 台北市長官邸, 台北, 台灣, 100

姓名: 賴清德

辦公室地址: 忠孝東路1段1號, 行政院, 台北市, 台灣, 100

送貨地址: 金華街142號, 行政院長官邸, 台北市, 台灣, 100

# 測試、執行

