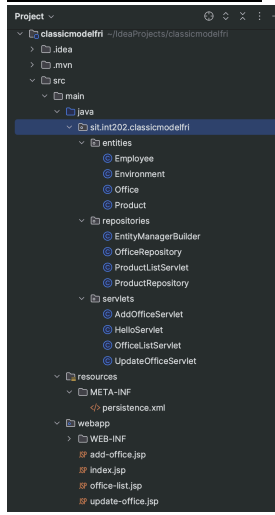
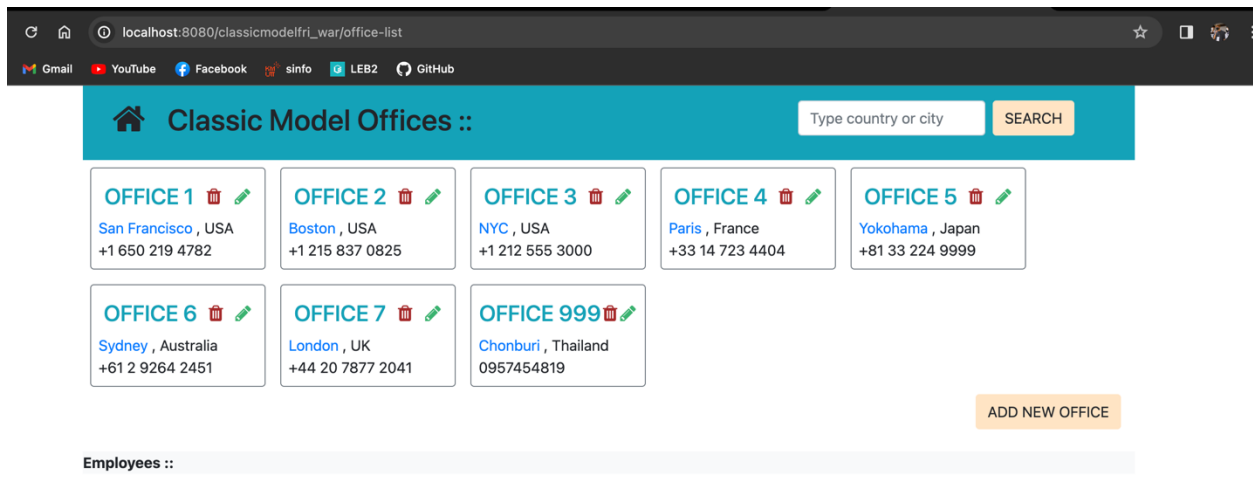


Project overview



Office list



ถ้ากดที่icon บัน จะสามารถกลับมาที่หน้า office-list ได้ และจะมีปุ่มที่สามารถsearch ด้วย country หรือ city ได้ ที่ด้านขวาบน และ officeแต่ละอันก็จะมี iconถังขยะ ที่จะสามารถลบofficeนั้นและ iconปากกา ที่จะสามารถแก้ไข ข้อมูลในofficeนั้นได้ และมี ปุ่ม ADD NEW OFFICEที่สามารถกดเพื่อเพิ่มofficeใหม่ได้

Method ที่เพิ่มมาใน OfficeRepository

validString() ซึ่งรับค่าเป็นString return ค่าเป็น Boolean ทำหน้าที่ในการเช็คค่า String ที่รับมานั้นเป็นnull หรือ empty string หรือไม่

2 usages

```
public boolean validString(String s){  
    if(s==null)return false;  
    if(s.trim().isEmpty())return false;  
    return true;  
}
```

validParam() ซึ่งรับค่าเป็นString ทำหน้าที่ในการเช็คค่าเป็นไปตาม validString() หรือไม่ ถ้าเป็นtrueให้ returnค่า String นั้น แต่ถ้าเป็นfalse ให้returnเป็นค่าnull

18 usages

```
public String validParam(String string){  
    if(validString(string)){  
        return string;  
    }return null;  
}
```

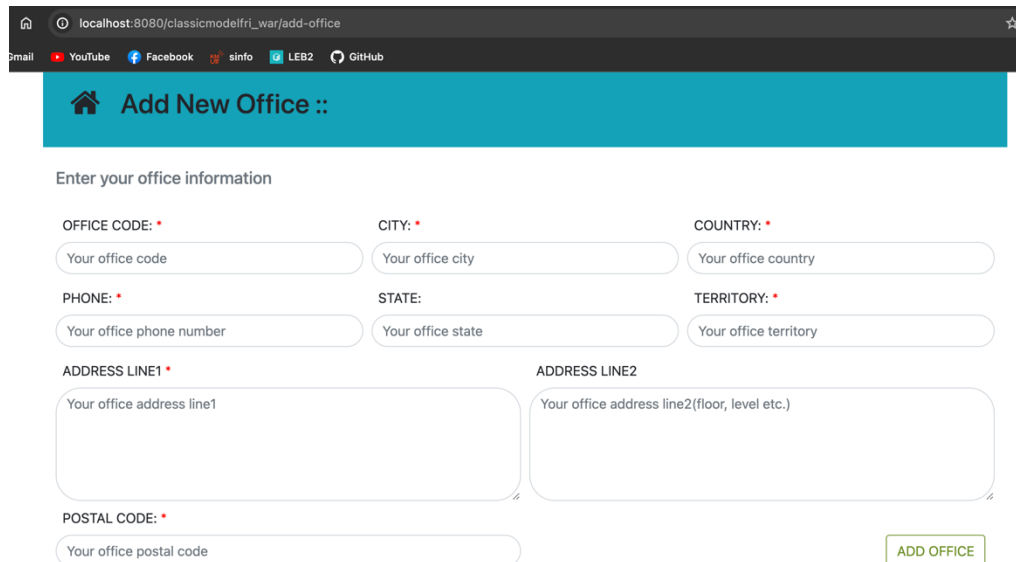
1.Insert(Add new office)

View:

```
JSP office-list.jsp x AddOfficeServlet.java JSP add-office.jsp OfficeListSer

<div class="col-12 d-flex flex-row-reverse">
  <div>
    <a href="add-office" class="btn btn-pink">ADD NEW OFFICE</a>
  </div>
</div>
</div>
<br>
```

เมื่อ user กดปุ่ม ADD NEW OFFICE จะไปถึงที่ add-office(servlet) ใน doGet() และใน doGet() ก็จะ redirect ไปที่ add-office.jsp โดยส่งต่อ request และ response ไป



localhost:8080/classicmodelfri_war/add-office

🏠 Add New Office ::

Enter your office information

OFFICE CODE: * CITY: * COUNTRY: *

Your office code Your office city Your office country

PHONE: * STATE: TERRITORY: *

Your office phone number Your office state Your office territory

ADDRESS LINE1 * ADDRESS LINE2

Your office address line1 Your office address line2(floor, level etc.)

POSTAL CODE: *

Your office postal code

ADD OFFICE

หน้า add-office.jsp จะมี form ให้ user กรอกข้อมูล และในการเก็บข้อมูลแต่ละช่องจะเก็บไว้ในตัวแปรที่แตกต่างกันออกไป เช่นการกรอกข้อมูล office code จะเก็บไว้ในตัวแปรชื่อ offcode และเมื่อ กด ADD OFFICE ค่าตัวแปรทั้งหมดจะถูกส่งไปที่ add-office(servlet) ใน doPost() แต่ถ้ากรอกข้อมูลที่มี * ไม่ครบทุกช่องจะไม่สามารถกดปุ่ม ADD OFFICE ได้

```

office-list.jsp  add-office.jsp  update-office.jsp
53  <form action="add-office" method="post">
54  <div class="form-row">
55  <div class="form-group col-md-4">
56  <label for="offCode" class="ml-2">OFFICE CODE: <span style="color:red">*</span></label>
57  <input type="text" name="offCode" class="form-control" id="offCode" placeholder="Your office code" required>
58  </div>
59  <div class="form-group col-md-4">
60  <label for="city" class="ml-2">CITY: <span style="color:red">*</span></label>
61  <input type="text" name="city" class="form-control" id="city" placeholder="Your office city" required>
62  </div>
63  <div class="form-group col-md-4">
64  <label for="country" class="ml-2">COUNTRY: <span style="color:red">*</span></label>
65  <input type="text" name="country" class="form-control" id="country" placeholder="Your office country" required>
66  </div>
67  </div>
68  <div class="form-row">
69  <div class="form-group col-md-4">
70  <label for="phone" class="ml-2">PHONE: <span style="color:red">*</span></label>
71  <input type="text" name="phone" class="form-control" id="phone" placeholder="Your office phone number" required>
72  </div>
73  <div class="form-group col-md-4">
74  <label for="state" class="ml-2">STATE: </label>
75  <input type="text" name="state" class="form-control" id="state" placeholder="Your office state">
76  </div>
77  <div class="form-group col-md-4">
78  <label for="territory" class="ml-2">TERRITORY: <span style="color:red">*</span></label>
79  <input type="text" name="territory" class="form-control" id="territory"
80  placeholder="Your office territory" required>
81  </div>
82  </div>
83  </form>

```

Controller:

```

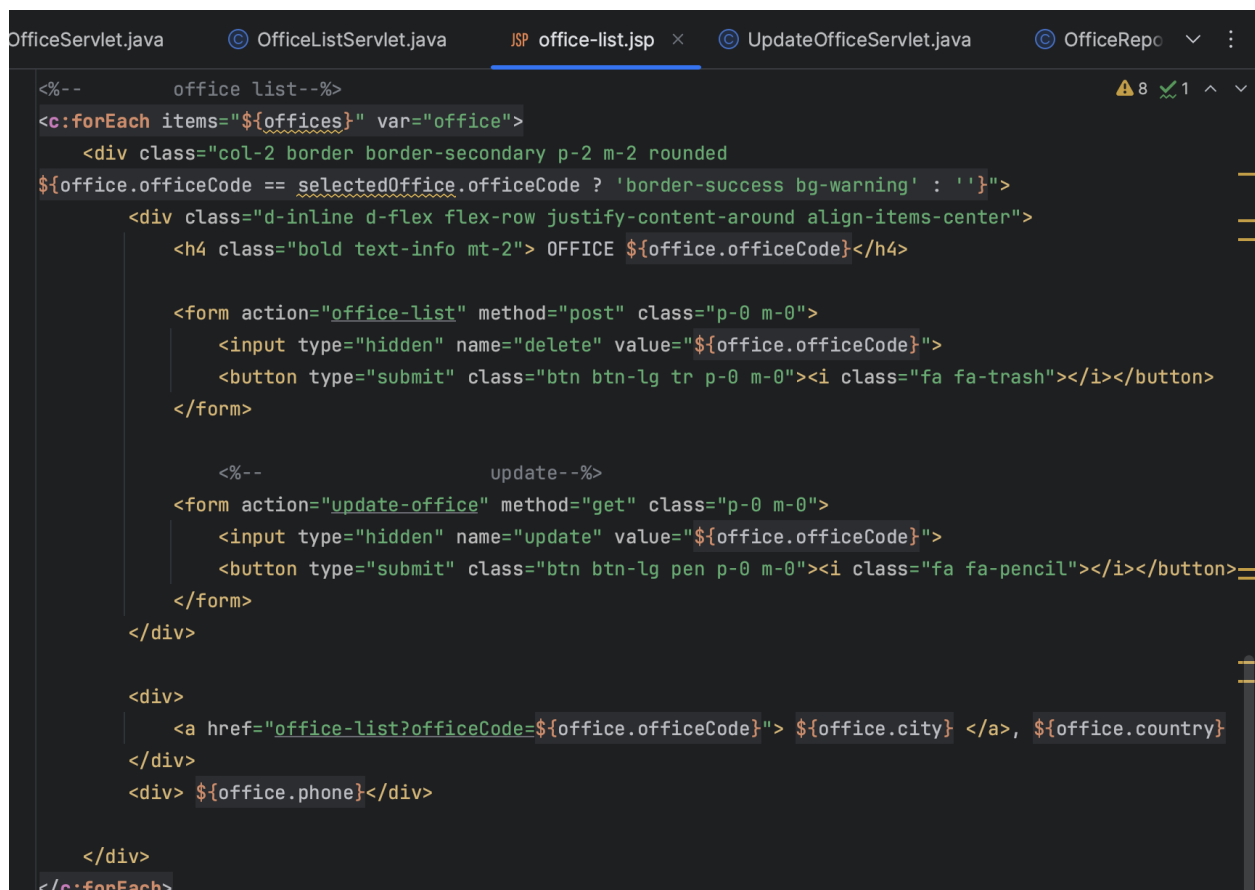
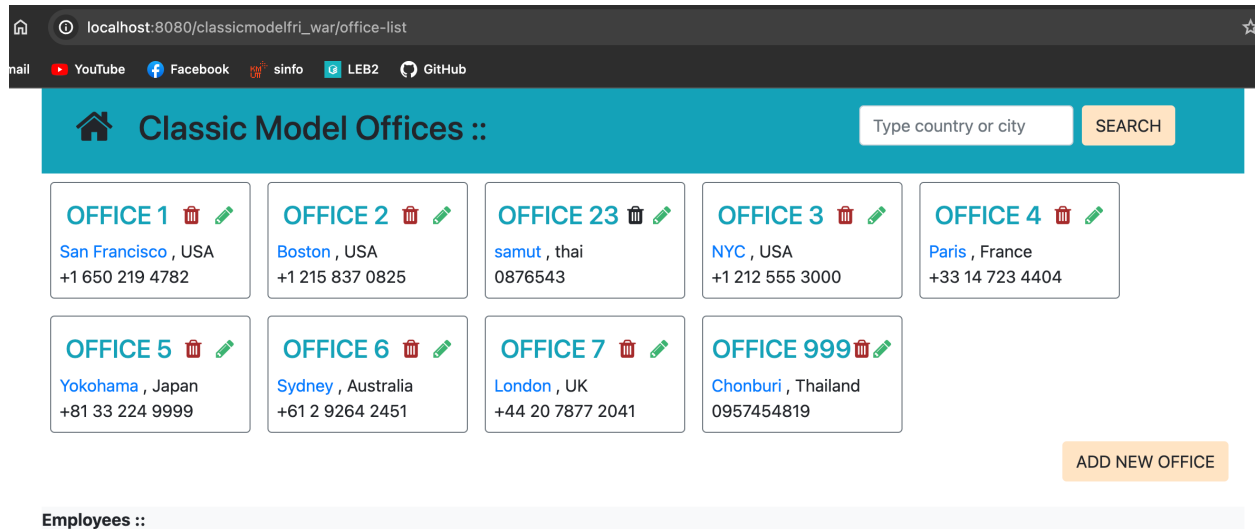
office-list.jsp  index.jsp  OfficeListServlet.java  Office.java  UpdateOfficeServlet.java  HelloServlet.java  AddOfficeServlet.java
17  no usages
18  @Override
19  @RequestMapping("/add-office")
20  protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
21  //get parameters from request from add-office.jsp
22  String officeCode = officeRepository.validParam(request.getParameter("offCode"));
23  String city = officeRepository.validParam(request.getParameter("city"));
24  String country = officeRepository.validParam(request.getParameter("country"));
25  String phone = officeRepository.validParam(request.getParameter("phone"));
26  String state = officeRepository.validParam(request.getParameter("state"));
27  String territory = officeRepository.validParam(request.getParameter("territory"));
28  String add1 = officeRepository.validParam(request.getParameter("address1"));
29  String add2 = officeRepository.validParam(request.getParameter("address2"));
30  String postCode = officeRepository.validParam(request.getParameter("postalCode"));
31
32
33  Office office = new Office();
34  //set the office data in insert into database
35  office.setOfficeCode(officeCode);
36  office.setCity(city);
37  office.setCountry(country);
38  office.setPhone(phone);
39  office.setState(state);
40  office.setTerritory(territory);
41  office.setAddressLine1(add1);
42  office.setAddressLine2(add2);
43  office.setPostalCode(postCode);
44
45  officeRepository.insert(office); //insert data into database
46  officeRepository.close();
47  response.sendRedirect(request.getContextPath() + "/office-list"); //redirect to the office-list
48
49  }
50  }

```

ในdoPost()จะสร้างofficeRepository เพื่อจัดการข้อมูลในdatabase และสร้างตัวแปรมาเพื่อรับค่า การกรอกข้อมูลของuser โดยเช็คผ่าน validParam()ในOfficeRepository และเมื่อได้ attribute ที่มีค่าที่userกรอกมาแล้ว ก็ใส่เข้าไปในoffice ตัวใหม่ ก่อนทำการinsert ลงในdatabase ผ่านการเรียกใช้ insert() ในOfficeRepository จากนั้นปิดการใช้งานofficeRepository แล้วจึงredirect กลับไปยังหน้า office-list(servlet)

2.DELETE(ถังขยะ)

View:



ปุ่ม delete จะเป็น icon รูปถังขยะที่อยู่ใน office แต่ละอัน เมื่อกดปุ่มถังขยะ จะทำการส่งตัวแปร delete ที่เก็บค่า office code ของ officenั้น และส่งต่อไปยัง doPost() ของ office-list(servlet)

Controller:

```
no usages
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    OfficeRepository officeRepository = new OfficeRepository();
    String officeCode = request.getParameter("delete");
    Office office = new Office();
    office.setOfficeCode(officeCode);

    officeRepository.delete(office);
    officeRepository.close();
    doGet(request, response);
}
}
```

doPost()เราจะสร้างตัวแปร officeCode มาเพื่อเก็บค่า delete และsetค่าofficeCodeลงในofficeที่สร้างขึ้นมาใหม่ และใช้ delete()ที่อยู่ใน OfficeRepository เพื่อลบofficeตัวนั้นในdatabase จากนั้นปิดการใช้งาน officeRepository แล้ว redirect กลับมาที่ doGet() ของ office-list

3.UPDATE(ปากกา)

View:

```
<%-- update--%>
<form action="update-office" method="get" class="p-0 m-0">
  <input type="hidden" name="update" value="${office.officeCode}">
  <button type="submit" class="btn btn-lg pen p-0 m-0"><i class="fa fa-pencil"></i></button>
</form>
</div>
```

เมื่อกดปุ่มรูปปากกาที่อยู่ในหน้าoffice-list จะเก็บค่าoffice code ในoffice ที่user กดแก้ไข ไว้ในตัวแปร update และส่งต่อไปยัง doGet() ใน update-office(servlet)

Controller:

```
@WebServlet(name = "UpdateOfficeServlet", value = "/update-office")
public class UpdateOfficeServlet extends HttpServlet {
    1 usage
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        OfficeRepository officeRepository = new OfficeRepository();
        String update = request.getParameter( s: "update");
        if (officeRepository.validString(update)){
            Office office = officeRepository.find(update);
            request.setAttribute( s: "office",office);
        }
        getServletContext().getRequestDispatcher( s: "/update-office.jsp").forward(request, response);
    }
}
```

doGet()เราจะสร้างofficeRepository มาเพื่อจัดการข้อมูลในdatabase และสร้างตัวแปรที่ชื่อว่าupdate ในการเก็บค่าupdate(office codeที่ส่งมา) และเช็คค่าupdate เป็นค่าnull หรือ empty stringหรือไม่ ผ่านการใช้ isValidString()ใน officeRepository เมื่อเช็คแล้วก็ใช้find()ในการค้นหาOffice ที่มีoffice code ตรงกับupdate แล้วเก็บOfficeที่ได้ลงในตัวแปร office และset office ลงในrequest และredirectกลับไปหน้า update-office.jsp

View:

Update Office ::

Enter your office information

OFFICE CODE: * CITY: * COUNTRY: *

1 San Francisco USA

PHONE: * STATE: TERRITORY: *

+1 650 219 4782 CA NA

ADDRESS LINE1 * ADDRESS LINE2

100 Market Your office address line2(floor, level etc.)

POSTAL CODE: *

94080

UPDATE OFFICE

```
t.jsp add-office.jsp update-office.jsp x
<form action="update-office" method="post">
  <div class="form-row">
    <div class="form-group col-md-4">
      <label for="offCode" class="ml-2">OFFICE CODE: <span style="color:red">*</span></label>
      <input type="text" name="offCode" class="form-control" id="offCode" readonly placeholder="Your office code" value="{office.offCode}" required>
    </div>
    <div class="form-group col-md-4">
      <label for="city" class="ml-2">CITY: <span style="color:red">*</span></label>
      <input type="text" name="city" class="form-control" id="city" placeholder="Your office city" value="{office.city}" required>
    </div>
    <div class="form-group col-md-4">
      <label for="country" class="ml-2">COUNTRY: <span style="color:red">*</span></label>
      <input type="text" name="country" class="form-control" id="country" placeholder="Your office country" value="{office.country}" required>
    </div>
  </div>
  <div class="form-row">
    <div class="form-group col-md-4">
      <label for="phone" class="ml-2">PHONE: <span style="color:red">*</span></label>
      <input type="text" name="phone" class="form-control" id="phone" placeholder="Your office phone number" value="{office.phone}" required>
    </div>
    <div class="form-group col-md-4">
      <label for="state" class="ml-2">STATE: </label>
      <input type="text" name="state" class="form-control" id="state" placeholder="Your office state" value="{office.state}">
    </div>
    <div class="form-group col-md-4">
      <label for="territory" class="ml-2">TERRITORY: <span style="color:red">*</span></label>
      <input type="text" name="territory" class="form-control" id="territory" placeholder="Your office territory" value="{office.territory}" required>
    </div>
  </div>
</div>
```

โดยในหน้า update-office.jsp จะมี form ที่มีค่า value เป็นค่าที่อยู่ในแต่ละ attributes ของ office นั้น แต่จะล็อกไม่ให้ user เปลี่ยนแปลงค่า officeCode ได้ และเมื่อ user กด ปุ่ม UPDATE OFFICE จะส่งค่าผ่านตัวแปรต่างๆ ไปที่ doPost() ของ update-office(servlet)

Controller:

```
office-list.jsp    JSP index.jsp    OfficeListServlet.java    Office.java    UpdateOfficeServlet.java    HelloServlet.java    AddOfficeServlet.java

getServletContext().getRequestDispatcher(s: "/update-office.jsp").forward(request, response);
}

no usages
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    OfficeRepository officeRepository = new OfficeRepository();
    String offCode = officeRepository.validParam(request.getParameter(s: "offCode"));
    String city = officeRepository.validParam(request.getParameter(s: "city"));
    String country = officeRepository.validParam(request.getParameter(s: "country"));
    String phone = officeRepository.validParam(request.getParameter(s: "phone"));
    String state = officeRepository.validParam(request.getParameter(s: "state"));
    String territory = officeRepository.validParam(request.getParameter(s: "territory"));
    String address1 = officeRepository.validParam(request.getParameter(s: "address1"));
    String address2 = officeRepository.validParam(request.getParameter(s: "address2"));
    String postalCode = officeRepository.validParam(request.getParameter(s: "postalCode"));

    Office newOffice = new Office();
    newOffice.setOfficeCode(offCode);
    newOffice.setCity(city);
    newOffice.setCountry(country);
    newOffice.setPhone(phone);
    newOffice.setState(state);
    newOffice.setTerritory(territory);
    newOffice.setAddressLine1(address1);
    newOffice.setAddressLine2(address2);
    newOffice.setPostalCode(postalCode);

    officeRepository.update(newOffice);
    officeRepository.close();
    response.sendRedirect(s: getServletContext().getContextPath()+"/office-list"); //ส่งผลลัพธ์แสดงออกมา
}
}
```

ใน `doPost()` เราจะสร้างตัวแปรที่เป็น `String` มาเพื่อรับค่าที่ user กรอกมา แล้วเราจะ `set` ค่าที่ user กรอกมา ในแต่ละ `attributes` ลงในตัวแปร `newOffice` ที่สร้างขึ้นมา และใช้ `update()` ที่อยู่ใน `officeRepository` ในการ `update` ข้อมูล `newOffice` ลงใน `database` จากนั้นปิดการใช้งาน `officeRepository` แล้วจึง `redirect` กลับไปที่ `office-list`

Result Update:

localhost:8080/classicmodelfri_war/update-office?update=1

YouTube


Facebook

sinfo

LEB2

GitHub

icons

 Update Office ::

Enter your office information

OFFICE CODE: *

CITY: *

COUNTRY: *

1

San Francisco

Thailand

PHONE: *

STATE:

TERRITORY: *

+1 650 219 4785

CA

NA

ADDRESS LINE1 *

ADDRESS LINE2

100 Market

Your office address line2(floor, level etc.)

POSTAL CODE: *

94080

UPDATE OFFICE

localhost:8080/classicmodelfri_war/office-list

mail

YouTube


Facebook

sinfo

LEB2



GitHub

icons



 Classic Model Offices ::

Type country or city



SEARCH

OFFICE 1  



San Francisco ,
Thailand
+1 650 219 4785

OFFICE 2  



Boston , USA
+1 215 837 0825

OFFICE 23  



samut , thai
+0876543

OFFICE 3  



NYC , USA
+1 212 555 3000

OFFICE 4  



Paris , France
+33 14 723 4404

OFFICE 5  

Yokohama , Japan
+81 33 224 9999

OFFICE 6  

Sydney , Australia
+61 2 9264 2451

OFFICE 7  

London , UK
+44 20 7877 2041

ADD NEW OFFICE

Employees ::

4.SEARCH BY COUNTRY OR CITY

View:

localhost:8080/classicmodelfri_war/office-list

Classic Model Offices ::

thai SEARCH

OFFICE 1 San Francisco , USA +1 650 219 4782	OFFICE 2 Boston , USA +1 215 837 0825	OFFICE 23 samut , thai 0876543	OFFICE 3 NYC , USA +1 212 555 3000	OFFICE 4 Paris , France +33 14 723 4404
OFFICE 5 Yokohama , Japan +81 33 224 9999	OFFICE 6 Sydney , Australia +61 2 9264 2451	OFFICE 7 London , UK +44 20 7877 2041	OFFICE 999 Chonburi , Thailand 0957454819	

ADD NEW OFFICE

Employees ::

```
<body>
<div class="container">
  <div class="row bg-info pt-2 pb-2">
    <div class="col-8 mt-2">
      <h2>
        <a href="office-list">
          <button class="btn btn-lg mt-0 pt-0" style="font-size: larger">
            <i class="fa fa-home"></i>
          </button>
        </a>
        Classic Model Offices ::
      </h2>
    </div>
    <div class="col-4 mt-2 mb-0 pb-0">
      <form action="office-list" method="get" class="form-inline">
        <input type="text" class="form-control" name="search" placeholder="Type country or city"
          value="{search}">
        <button type="submit" class="btn btn-pink ml-2">SEARCH</button>
      </form>
    </div>
  </div>
</div>
```

เมื่อ user กรอกข้อมูลและ กดปุ่ม search จะทำการส่ง ตัวแปรsearch ที่มีค่าเป็น stringที่userกรอกมา ไปที่หน้า office-list(servlet) ในmethod doGet() และมี valueเป็นตัวแปรsearch และเมื่อกดicon บ้าน ซ้ายบนจะลี้้งกลับไป office-list(servlet) ใน doGet() และในdoGet() ก็จะ Redirect กลับมาที่หน้า office-list.jsp

Controller:

```
@WebServlet(name = "OfficeListServlet", value = "/office-list")
public class OfficeListServlet extends HttpServlet {

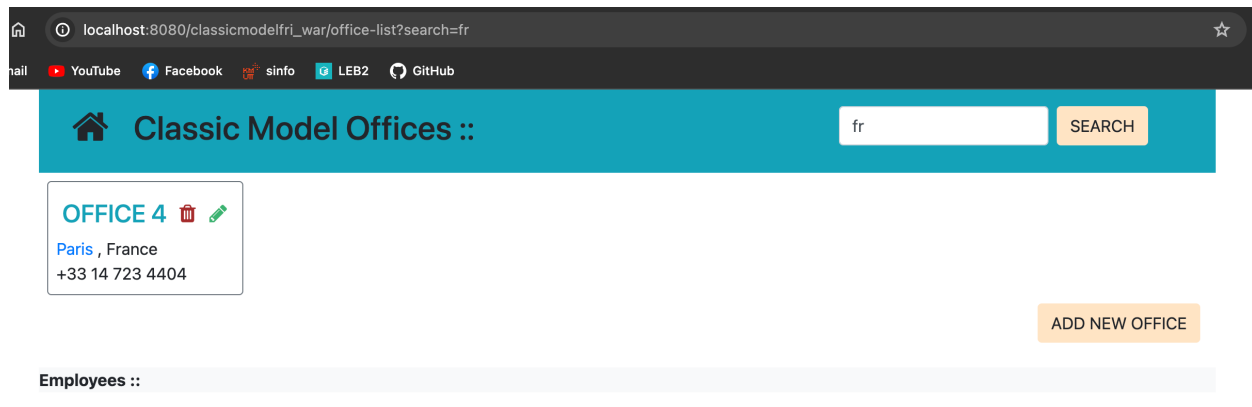
    1 usage
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        OfficeRepository officeRepository = new OfficeRepository();
        request.setAttribute(s: "offices", officeRepository.findAll());
        String officeCode = request.getParameter(s: "officeCode");
        String search = request.getParameter(s: "search");
        if(officeRepository.validString(search)){
            request.setAttribute(s: "search",search);
            request.setAttribute(s: "offices",officeRepository.findByCityOrCountry(search));
        }
        if (officeCode != null) {
            request.setAttribute(s: "selectedOffice", officeRepository.find(officeCode));
        }

        getServletContext().getRequestDispatcher(s: "/office-list.jsp").forward(request, response);
    }
}
```

doGet() เราจะสร้างตัวแปรมาเพื่อรับค่า search ที่userกรอกมา แล้วนำค่านี้ไปเช็คใน validString() โดยถ้าไม่เป็น null และ empty string จะให้ set ค่า search เพื่อส่งกลับไปให้ยังแสดงอยู่ในช่อง search bar และ นำค่า search ไปใช้ใน findByCityOrCountry() เพื่อหา office ที่ตรงกับ search แล้ว setค่าoffices ลงใน request แล้ว redirect กลับไปที่หน้า office-list.jsp

Result search by county or city:

The screenshot shows a web browser at localhost:8080/classicmodelfri_war/office-list?search=yokohama. The page has a teal header with a home icon, the text 'Classic Model Offices ::', a search input field containing 'yokohama', and a 'SEARCH' button. Below the header, there is a card for 'OFFICE 5' with a trash icon and a pencil icon. The card text reads 'Yokohama , Japan' and '+81 33 224 9999'. To the right of the card is an 'ADD NEW OFFICE' button. At the bottom, there is a section titled 'Employees ::' followed by a horizontal line.



กดที่icon บ้าน

