1.Insert

Controller

doGet

getServletContext().getRequestDispatcher("/office_insert.jsp").forward(request, response); ส่งต่อไปที่ office_list.jsp (office-list)

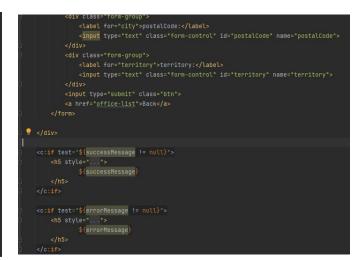
doPost

สร้าง Object **"Office"** จาก Office และดึงค่าของพารามิเตอร์ officeCode, city, phone, addressLine1, country, postalCode, territory ที่ User ส่งมาเก็บไว้ใน String

ทำการเช็กเงื่อนไขถ้าพารามิเตอร์ไม่มีค่าส่งมา ให้ set attribute "errorMessage" ข้อความแสดง "!! Insert failed!! " ใน request จากนั้น set ค่าที่รับมาจากพารามิเตอร์ลง "Office"

ทำการ instance OfficeRepository เพื่อใช้คำสั่งในการ insert ข้อมูลในฐานข้อมูล และเช็คเงื่อนไข officeRepository.insert (office) จะถูกเรียกเพื่อเพิ่มข้อมูล office แล้ว set attribute "successMessage" ข้อความแสดง "!! Insert successfully !!" ใน request ถ้าไม่ตรงเงื่อนไข จะ set attribute ข้อความแสดง "!! Insert failed !!" และส่งต่อไปที่ doGet

Insert : ปุ่ม link ไปที่ office list.jsp (office-list) และ ไปที่ method doGet

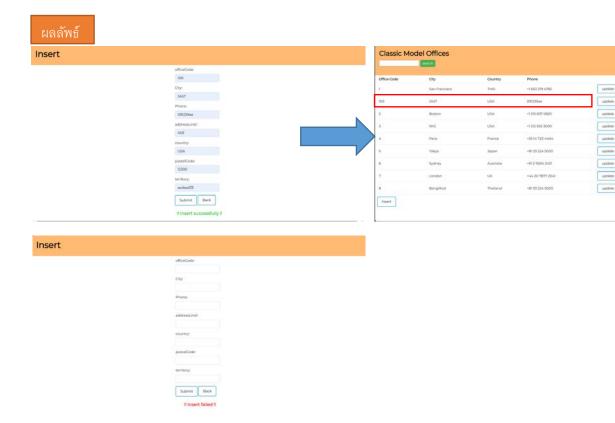


delete

delete

ทำ form รับค่าที่ User กรอกเข้ามาเพื่อทำการ Insert ข้อมูล

ใช้ JSTL ในการเช็กเงื่อนใข ถ้า "successMessage" ไม่เท่ากับ null จะแสดงข้อความ "!! Insert successfully !!" และถ้า "errorMessage" ไม่เท่ากับ null จะแสดงข้อความ "!! Insert failed!!"



2.Delete

Controller

doGet

ดึงค่าของพารามิเตอร์ officeCode จากคำขอ HTTP ที่ User กรอกเข้ามา จากนั้นทำการ instance OfficeRepository เพื่อใช้ คำสั่งในการ delete ข้อมูลในฐานข้อมูล ซึ่งการ delete จะใช้ officeCode ในการไปลบข้อมูล ส่วนการเช็คเงื่อนไข officeRepository.delete(officeCode) จะถูกเรียกเพื่อลบ office โดยใช้รหัส officeCode ที่ได้มา แล้ว set attribute "successMessage" ข้อความแสดง "Delete successfully!!" ใน request ถ้าไม่ตรงเงื่อนไข จะ set attribute "errorMessage" ข้อความแสดง "Delete failed!!"

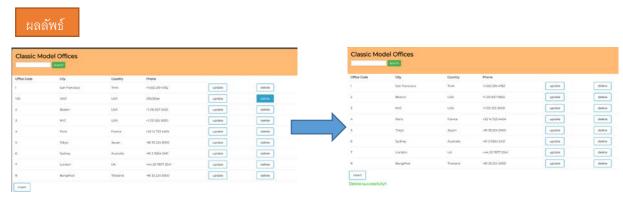
request.getRequestDispatcher("office-list").forward(request, response); จากนั้น ส่งต่อไปที่ office_list.jsp (office-list)

View

delete

delete : ทำ hyperlink ในการถบ office และจะส่ง officeCode ไปที่ servlet doGet

ใช้ JSTL ในการเช็คเงื่อนไข ถ้า "successMessage" ไม่เท่ากับ null จะแสดงข้อความ "Delete successfully!!" และถ้า "errorMessage" ไม่เท่ากับ null จะแสดงข้อความ "Delete failed!!"



Controller

Office Code	City	Country	Phone		
	San Proncisco	THE	+1 650 219 4762	update	delate
100	SAST	USA	09/259ee	update	delete
	Boston	USA	+1 215 837 0825	update	delete
	MYC	USA	+1 212 558 3000	update	delete
4	Daris	France	+33 14 723 4404	update	delete
y .	Tokyo	Japan	+81 53 324 5000	update	delete
0.0	Sydney	Austrelia	+612 9264 2453	Supdate	delete
7	London	UK	+64-20 7877 2041	tapidate	delete
8	BengMod	Theland	-61 33 224 5000	update	delete

3.Update

```
package stt.int202.classicmodels.controllers;

@MebServlet(name = "OfficeUpdateServlet", value = "/office-update")
public class OfficeUpdateServlet extends HttpServlet {
    lurage
    @Override

protected void doSet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String officeCode = request.getParameter( @ "officeCode");
    OfficeRepository officeRepository friedRepository();
    request.setAttribute( @ "selected", officeRepository.find(officeCode));
    getServletContext().getRequestDispatcher( @ "/office_update.jsp").forward(request, response);
}
```

doGet

ดึงค่าของพารามิเตอร์ officeCode จากคำขอ HTTP ที่ User กรอกเข้ามา จากนั้นทำการ instance OfficeRepository เพื่อใช้ คำสั่งในการ find ข้อมูลในฐานข้อมูล แล้วนำไป set attribute ว่า "selected" จากนั้น ส่งต่อไปที่ office_update.jsp (office-update)

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, 10Exception {
    Office newOffice = new Office():
        String officeCode = request.getParameter( * 'officeCode');
        String officeCode = request.getParameter( * 'officeCode');
        String addressLine1 = request.getParameter( * 'addressLine1');
        String addressLine2 = request.getParameter( * 'addressLine2');
        String addressLine2 = request.getParameter( * 'addressLine2');
        String addressLine2 = request.getParameter( * 'addressLine2');
        String country = request.getParameter( * 'country');
        String postalCode = request.getParameter( * 'country');
        String postalCode = request.getParameter( * 'retritory');
        nemOffice.setOfficeCode():
        newOffice.setOfficeCode():
        newOffice.setOfficeCode():
        newOffice.setCotty(city);
        newOffice.setCotty(city);
        newOffice.setCotty(city);
        newOffice.setCountry(country);
        newOffice.set
```

doPost

สร้าง Object "newOffice" จาก Office และดึงค่าของพารามิเตอร์ officeCode, city, phone, addressLine1, addressLine2, state, country, postalCode, territory ที่ User ส่งมาเก็บไว้ใน String แล้ว set ค่าที่รับมาจาก พารามิเตอร์ลง "newOffice" จากนั้นทำการ instance OfficeRepository เพื่อใช้คำสั่งในการ update ข้อมูลในฐานข้อมูล และส่งต่อไปที่ office_update.jsp (office-update)

```
reity class="form-group">

div class="form-group">

diput type="fort class="form-control" id="phone" name="phone" value="$[melected_phone]">

diput type="fort" class="form-control" id="saddressline2" name="dodressline2" value="$[melected_addressline2]">

diput type="fort" class="form-control" id="saddressline2" name="dodressline2" value="$[melected_addressline2]">

diput type="fort" class="form-control" id="state" name="state" value="$[melected_state]">

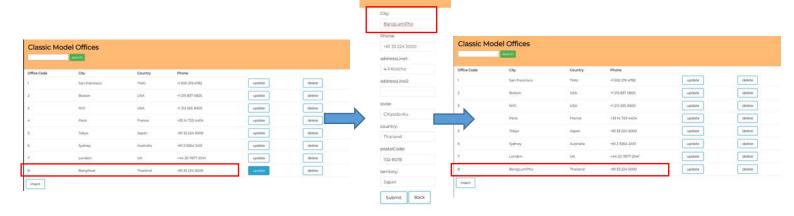
diput type="fort" class="form-control"
```

ทำ form เพื่อรับค่าพารามิเตอร์ที่ User กรอกเข้ามาเพื่อจะทำการเปลี่ยนแปลงข้อมูล

<input type="hidden" name="officeCode" value="\${selected.officeCode}">: ดึงค่าของ "officeCode" จาก attribute "selected" ใช้เพื่อให้เห็นข้อมูลเดิมก่อนจะทำการเปลี่ยนแปลงข้อมูล (type hidden ช่องป้อนข้อมูลจะไม่ขึ้นหน้าเว็บแต่ยังคง เป็นส่วนหนึ่งของแบบฟอร์มและสามารถส่งพร้อมกับข้อมูลแบบฟอร์มอื่นๆ)

update : ทำ hyperlink ในการ update office และจะส่ง officeCode ไปที่ servlet doGet





```
4. AUMINIBLE city, country

no usages
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    OfficeRepository officeRepository = new OfficeRepository();
    String find = request.getParameter( & "find");
    request.setAttribute( & "offices", officeRepository.findByCityOrCountry(find));
    getServletContext().getRequestDispatcher( & "/office_list.jsp").forward(request, response);
}
```

doPost

ทำการ instance OfficeRepository เพื่อใช้คำสั่งในการ find ข้อมูลในฐานข้อมูล แล้วนำไป set attribute ว่า "offices" และคึง ค่าของพารามิเตอร์ที่ User ต้องการหาลงใน "find" จากนั้นส่งต่อไปที่ office_list.jsp (office-list)

ทำ form รับค่าพารามิเตอร์ที่ User กรอกเข้ามา เพื่อทำการ find city หรือ contry กด search จะส่งไปที่ doPost servlet

