

OfficeDeleteServlet

```
package sit.int202.classicmodels.servlet;

import ...

@WebServlet(name = "OfficeDeleteServlet", value = "/office-delete")
public class OfficeDeleteServlet extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        String officeCode = request.getParameter("officeCode");
        OfficeRepository officeRepository = new OfficeRepository();
        if (officeRepository.delete(officeCode)) {
            request.setAttribute("successMessage", "Delete office successfully");
        } else {
            request.setAttribute("errorMessage", "Delete office failed");
        }
        request.getRequestDispatcher("office-list").forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    }
}
```

Servlet นี้เมื่อถูกเข้ามา จะเข้าไปทำงานที่ `doGet` ซึ่งจะ request `officeCode` มาเพื่อใช้ในการ delete โดยถ้าผ่านเงื่อนไขก็จะไป set ตัวแปร `successMessage` ให้เป็นข้อความที่ตั้งไว้ ถ้าไม่ก็จะไป set `errorMessage`

OfficeInsertServlet

```
20 @Override
21 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
22     Office office = new Office();
23     String officeCode = request.getParameter("officeCode");
24     String city = request.getParameter("city");
25     String phone = request.getParameter("phone");
26     String addressLine1 = request.getParameter("addressLine1");
27     String country = request.getParameter("country");
28     String postalCode = request.getParameter("postalCode");
29     String territory = request.getParameter("territory");
30
31     office.setOfficeCode(officeCode);
32     office.setCity(city);
33     office.setPhone(phone);
34     office.setAddressLine1(addressLine1);
35     office.setCountry(country);
36     office.setPostalCode(postalCode);
37     office.setTerritory(territory);
38
39     OfficeRepository officeRepository = new OfficeRepository();
40
41     if (officeRepository.insert(office)) {
42         request.setAttribute("successMessage", "Insert office successfully");
43     } else {
44         request.setAttribute("errorMessage", "Insert office failed");
45     }
46 }
```

จะทำงานเมื่อ user ใส่ข้อมูลใน form ที่ office_insert.jsp โดยจะไป set value ต่างๆ ซึ่งถ้าใส่ถูกต้องตามเงื่อนไขก็ set ตัวแปร successMessage ให้เป็นข้อความที่ตั้งไว้ ถ้าไม่ก็จะไป set errorMessage เพื่อแสดงบอก user ว่าสำเร็จหรือไม่

OfficeSearchServlet

```
@WebServlet(name = "OfficeSearchServlet", value = "/search-office")
public class OfficeSearchServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        OfficeRepository officeRepository = new OfficeRepository();
        String find = request.getParameter("find");
        request.setAttribute("offices", officeRepository.findByCityOrCountry(find));

        getServletContext().getRequestDispatcher("/office_list.jsp").forward(request, response);
    }
}
```

servletนี้จะทำงานเมื่อ user ใส่ตัวอักษรลงไปใน search form แล้ว โดยจะrequest find เพื่อไปsetAttribute ให้officelist ขึ้นแต่อันที่find

doGet OfficeUpdateServlet

```
@WebServlet(name = "OfficeUpdateServlet", value = "/office-update")
public class OfficeUpdateServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        String officeCode = request.getParameter("officeCode");
        OfficeRepository officeRepository = new OfficeRepository();
        request.setAttribute("selected", officeRepository.find(officeCode));
        getServletContext().getRequestDispatcher("/office_update.jsp").forward(request, response);
    }
}
```

servlet update doGet เมื่อ user เข้ามาในหน้า update โดยจะทำการrequest parameter officecodeเพื่อใช้เป็น ตัวบอกว่าจะupdate ที่ตัวไหน

doPost OfficeUpdateServlet

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    Office newOffice = new Office();
    String officeCode = request.getParameter("officeCode");
    String city = request.getParameter("city");
    String phone = request.getParameter("phone");
    String addressLine1 = request.getParameter("addressLine1");
    String addressLine2 = request.getParameter("addressLine2");
    String state = request.getParameter("state");
    String country = request.getParameter("country");
    String postalCode = request.getParameter("postalCode");
    String territory = request.getParameter("territory");

    newOffice.setOfficeCode(officeCode);
    newOffice.setCity(city);
    newOffice.setPhone(phone);
    newOffice.setAddressLine1(addressLine1);
    newOffice.setAddressLine2(addressLine2);
    newOffice.setState(state);
    newOffice.setCountry(country);
    newOffice.setPostalCode(postalCode);
    newOffice.setTerritory(territory);

    OfficeRepository officeRepository = new OfficeRepository();
    officeRepository.update(newOffice);
    getServletContext().getRequestDispatcher("/office_update.jsp").forward(request, response);
}
```

ทำงานเมื่อ user ใสข้อมูลใน form ลงไปเสร็จแล้วก็จะrequest parameter ทั้งหมดที่กรอกไป เพื่อจะ set value ให้แต่ละอัน

OfficeRepository insert

```
1 usage  ± unknown *
public boolean insert(Office office) {
    try {
        EntityManager entityManager = getEntityManager();
        entityManager.getTransaction().begin();
        entityManager.persist(office);
        entityManager.getTransaction().commit();
    } catch (Exception e) {
        System.out.println();
        return false;
    }
    return true;
}
```

Try catch เพื่อที่จะไม่ให้มัน error หาก error print () เพื่อเว็บจะได้ไม่ล่ม

OfficeRepository delete

```
2 usages  ± unknown
public boolean delete(String officeCode) {
    EntityManager entityManager = getEntityManager();
    Office office = find(officeCode);
    try {
        if (office != null) {
            entityManager.getTransaction().begin();
            entityManager.remove(office);
            entityManager.getTransaction().commit();
        }
        return true;
    } catch (Exception e) {
        System.out.println(e);
        return false;
    }
}
```

Try catch เพื่อที่จะไม่ให้มัน error หาก error print () เพื่อเว็บจะได้ไม่ล่ม

รับ entity มาแล้วหา officeCode จากนั้น remove office ที่ find มา

OfficeRepository update

```
1 usage  ± unknown
public boolean update(Office newOffice) {
    if (newOffice != null) {
        EntityManager entityManager = getEntityManager();
        Office office = find(newOffice.getOfficeCode());
        if (office != null) {
            entityManager.getTransaction().begin();
            office.setCity(newOffice.getCity());
            office.setPhone(newOffice.getPhone());
            office.setAddressLine1(newOffice.getAddressLine1());
            office.setAddressLine2(newOffice.getAddressLine2());
            office.setState(newOffice.getState());
            office.setCountry(newOffice.getCountry());
            office.setPostalCode(newOffice.getPostalCode());
            office.setTerritory(newOffice.getTerritory());
            entityManager.getTransaction().commit();
            return true;
        }
    }
    return false;
}
```

ถ้าใส่ข้อมูลเข้ามา ก็จะได้รับค่ามาแล้ว find officeCode เพื่อเจาะจง office ที่จะ update แล้วทำการ set ค่าทั้งหมด

office_insert.jsp

```
<div class="form-group">
  <label for="phone">country:</label>
  <input type="text" class="form-control" id="country" name="country" >
</div>
<div class="form-group">
  <label for="city">postalCode:</label>
  <input type="text" class="form-control" id="postalCode" name="postalCode">
</div>
<div class="form-group">
  <label for="territory">territory:</label>
  <input type="text" class="form-control" id="territory" name="territory" >
</div>
<input type="submit" value="Submit" />
<a href="office-list">Back</a>
<c:if test="${successMessage != null}">
  <div style="background-color: #d4edda; padding: 5px; margin-bottom: 10px;">
    <div style="display: flex; align-items: center;">
      <div style="width: 20px; height: 20px; background-color: #28a745; border-radius: 50%; margin-right: 10px;">
```

สร้างform insert-office แล้วทำการเรียกใช้ successMessage และ errorMessage ในบอกuserว่าสามารถ insert ได้หรือไม่

office_list.jsp

[illegible]

สร้าง table ด้วย foreach โดยให้เรียก item ไปที่ office entities

โดยถ้าเรา delete จะเห็นได้ว่า success หรือ error ซึ่งมาจากใน OfficeDeleteServlet

โดยถ้าsuccessMessage ไม่ใช่ nullจะขึ้นmessageที่ตั้งไว้

Classic Model Offices ::				find
Office Code	City	Country	Phone	
1	San Francisco	USA	+1 650 219 4782	update delete
2	Boston	USA	+1 215 837 0825	update delete
3	NYC	USA	+1 212 555 3000	update delete
4	Paris	France	+33 14 723 4404	update delete
5	Tokyo	Japan	+81 33 224 5000	update delete
6	Sydney	Australia	+61 2 9264 2451	update delete
7	London	UK	+44 20 7877 2041	update delete

Classic Model Offices ::

find

Office Code		City	Country	Phone	Delete office failed	
					update	delete
1		San Francisco	USA	+1 650 219 4782	update	delete
2		Boston	USA	+1 215 837 0825	update	delete
3		NYC	USA	+1 212 555 3000	update	delete
4		Paris	France	+33 14 723 4404	update	delete
5		Tokyo	Japan	+81 33 224 5000	update	delete
6		Sydney	Australia	+61 2 9264 2451	update	delete
7		London	UK	+44 20 7877 2041	update	delete

Insert

Delete Failed

Classic Model Offices ::

find

Office Code		City	Country	Phone	Delete office successfully	
					update	delete
1		San Francisco	USA	+1 650 219 4782	update	delete
2		Boston	USA	+1 215 837 0825	update	delete
3		NYC	USA	+1 212 555 3000	update	delete
4		Paris	France	+33 14 723 4404	update	delete
5		Tokyo	Japan	+81 33 224 5000	update	delete
6		Sydney	Australia	+61 2 9264 2451	update	delete
7		London	UK	+44 20 7877 2041	update	delete

Insert

Delete Success

offics_update.jsp

```
<div class="container">
  <div class="row bg-primary">
    <h2>Update ::</h2>
  </div>
  <div class="row">
    <form method="post" action="office-update">
      <div class="form-group">
        <label for="city">City:</label>
        <input type="hidden" name="officeCode" value="${selected.officeCode}">
        <input type="text" class="form-control" id="city" name="city" value="${selected.city}">
      </div>
      <div class="form-group">
        <label for="phone">Phone:</label>
        <input type="text" class="form-control" id="phone" name="phone" value="${selected.phone}">
      </div>
      <div class="form-group">
        <label for="addressLine1">addressLine1:</label>
        <input type="text" class="form-control" id="addressLine1" name="addressLine1" value="${selected.addressLine1}">
      </div>
      <div class="form-group">
        <label for="addressLine2">addressLine2:</label>
        <input type="text" class="form-control" id="addressLine2" name="addressLine2" value="${selected.addressLine2}">
      </div>
    </form>
  </div>
</div>
```

format เหมือนกับinsertแต่จะขึ้นvalueที่ update อยู่

Update ::

City:

Phone:

addressLine1:

addressLine2:

state:

country:

postalCode:

territory:

[Back](#)