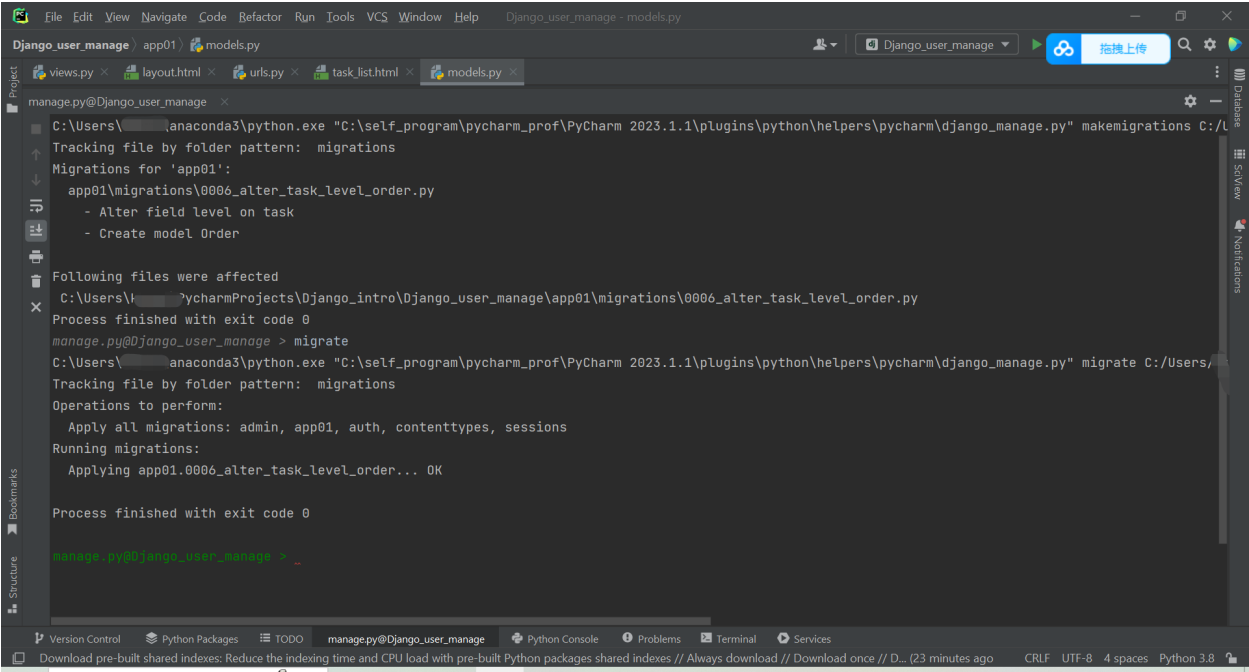


ajax in js

1. create table

```
class Order(models.Model):
    oid = models.CharField(verbose_name="Order ID", max_length=64)
    title = models.CharField(verbose_name="Name", max_length=64)
    price = models.IntegerField(verbose_name='Price')
    status_choice = {
        (1, "paid"),
        (2, "unpaid"),
    }
    status = models.SmallIntegerField(verbose_name="Status", choices=status_choice, default=1)
    admin = models.ForeignKey(verbose_name='Administrator', to="Admin", on_delete=models.CASCADE)
```



2. add modal

previous idea:

```
connection

button: data-toggle="modal" data-target="#myModal"
modal: id="myModal"
```

```
{% extends 'layout.html' %}

{% block content %}

    <div class="container">
        <div>
            <input data-toggle="modal" data-target="#myModal" type="button" value="New Order" class="btn btn-primary">
        </div>

        <!-- Modal -->
        <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
                            aria-hidden="true">&times;</span></button>
                        <h4 class="modal-title" id="myModalLabel">Modal title</h4>
                    </div>
                    <div class="modal-body">
                        ...
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
                        <button type="button" class="btn btn-primary">Save changes</button>
                    </div>
                </div>
            </div>
        </div>

    </div>
```

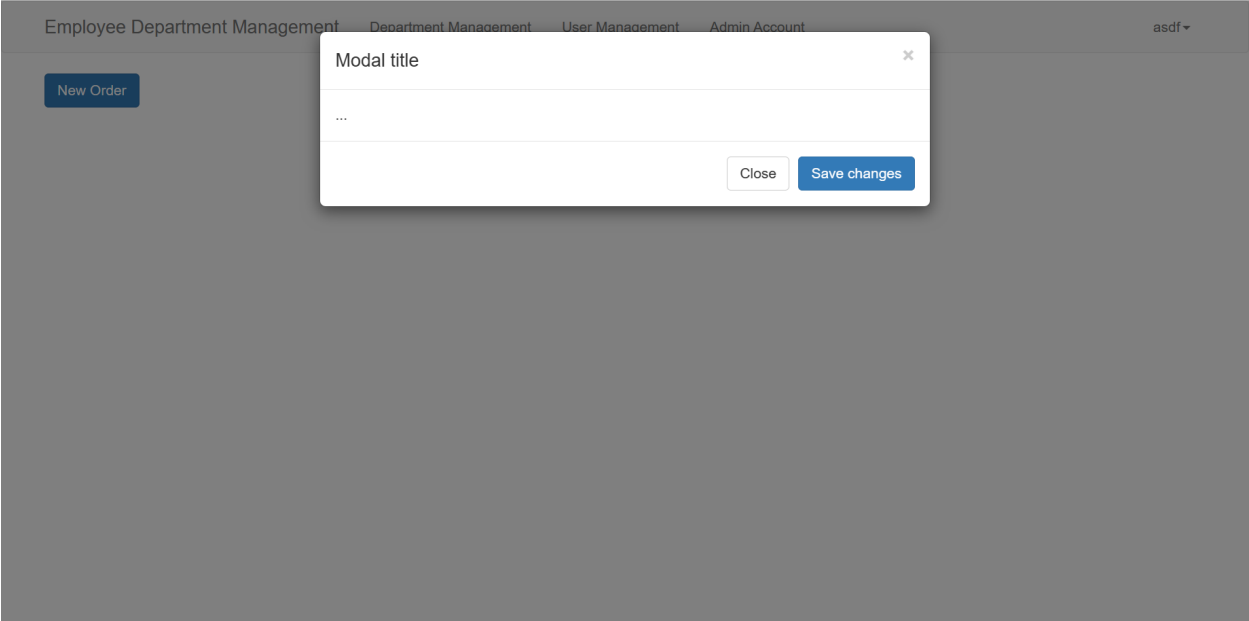
```

        </div>
      </div>
    </div>

{% endblock %}

{% block js %}
{% endblock %}

```



using js

```

{% extends 'layout.html' %}

{% block content %}

    <div class="container">
        <div>
            <input id="btnAdd" type="button" value="New Order" class="btn btn-primary">
        </div>

        <!-- Modal -->
        <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
                            aria-hidden="true">&times;</span></button>
                        <h4 class="modal-title" id="myModalLabel">Modal title</h4>
                    </div>
                    <div class="modal-body">
                        ...
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
                        <button type="button" class="btn btn-primary">Save changes</button>
                    </div>
                </div>
            </div>
        </div>

    </div>

{% endblock %}
{% block js %}
<script>
    $(function () {
        bindBtnAddEvent();
    })

    function bindBtnAddEvent() {
        $('#btnAdd').click(function () {
            $('#myModal').modal('show');
        });
    }
</script>
{% endblock %}

```

方法

`.modal(options)`

将页面中的某块内容作为模态框激活。接受可选参数 `object`。

`$('#myModal').modal({
 keyboard: false
});`

Copy

`.modal('toggle')`

手动打开或关闭模态框。在模态框显示或隐藏之前返回到主调函数中（也就是，在触发 `shown.bs.modal` 或 `hidden.bs.modal` 事件之前）。

`$('#myModal').modal('toggle')`

Copy

`.modal('show')`

手动打开模态框。在模态框显示之前返回到主调函数中（也就是，在触发 `shown.bs.modal` 事件之前）。

`$('#myModal').modal('show')`

Copy

`.modal('hide')`

手动隐藏模态框。在模态框隐藏之前返回到主调函数中（也就是，在触发 `hidden.bs.modal` 事件之前）。

`$('#myModal').modal('hide')`

Copy

概览

过渡效果

模态框

实例

Sizes

Remove animation

Varying content based on trigger button

用法

参数

方法

事件

下拉菜单

滚动监听

标签页

工具提示

弹出框

警告框

按钮

Collapse

Carousel

Affix

返回顶部

JavaScript 插件 · Bootstrap v3 中文文档 | Bootstrap 中文网

jQuery 插件为 Bootstrap 的组件赋予了“生命”。可以简单地一次性引入所有插件，或者逐个引入到你的页面中。

B

<https://v3.bootcss.com/javascript/#modals>

3. add form in modal

if modal cannot show, when using js. Please first check <script> part is in js block or not.

```
{% extends 'layout.html' %}

{% block content %}

<div class="container">
  <div>
    <input data-toggle="modal" data-target="#myModal" type="button" value="New Order Button 1"
      class="btn btn-primary">
    <input id="btnAdd" type="button" value="New Order Button 2" class="btn btn-primary">
  </div>
  <!-- Modal -->
  <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span></button>
          <h4 class="modal-title" id="myModalLabel">Modal title</h4>
        </div>
        <div class="modal-body">
          <form id="formAdd">
            <div class="clearfix">
              {% for field in form %}
                <div class="col-xs-6">
                  <div class="form-group" style="position: relative">
                    <label>{{ field.label }}</label>
                    {{ field }}
                    <span class="error-message" style="color: red; position: absolute"></span>
                  </div>
                </div>
              {% endfor %}
            </div>
          </form>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
          <button type="button" class="btn btn-primary">Save changes</button>
        </div>
      </div>
    </div>
  </div>

</div>

{% endblock %}
{% block js %}
<script type="text/javascript">
  $(function () {
    bindBtnAddEvent();
  })
</script>
</block>
```

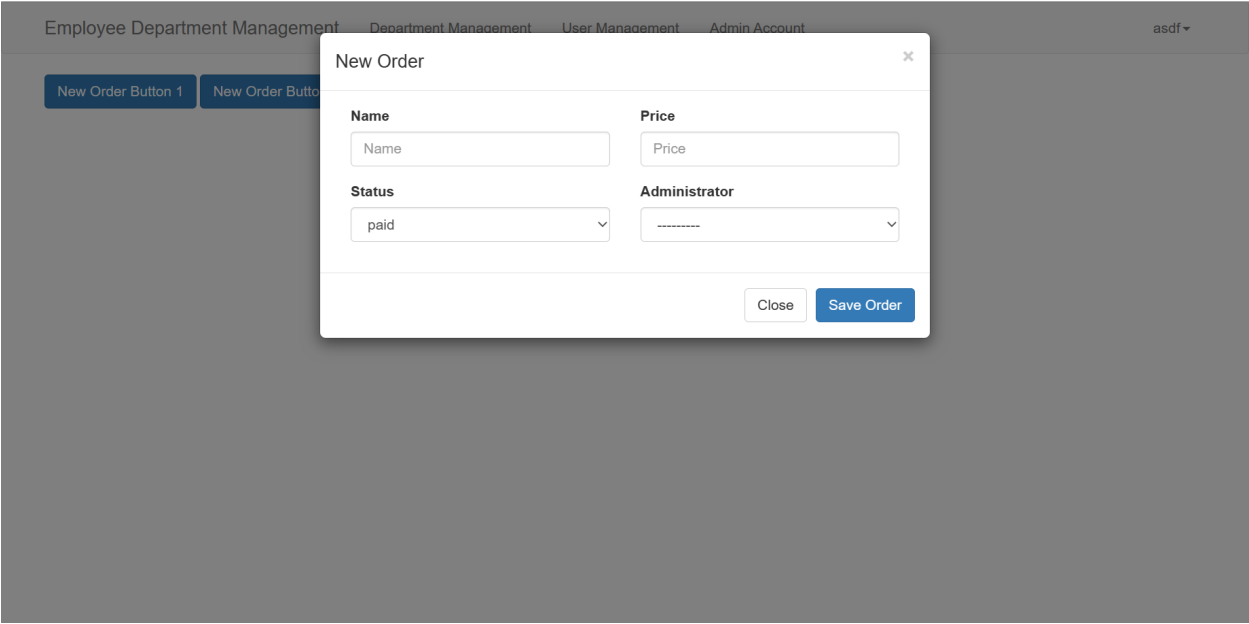
ajax in js

3

```
function bindBtnAddEvent() {
    $('#btnAdd').click(function () {
        $('#myModal').modal('show');
    });

}
</script>

{% endblock %}
```



4. save order

```
@csrf_exempt
def order_add(request):

    form = OrderModelForm(data=request.POST)
    if form.is_valid():
        form.save()
        return HttpResponse('^_^')

    return HttpResponse('^_!')
```

order id

first exclude it in input form

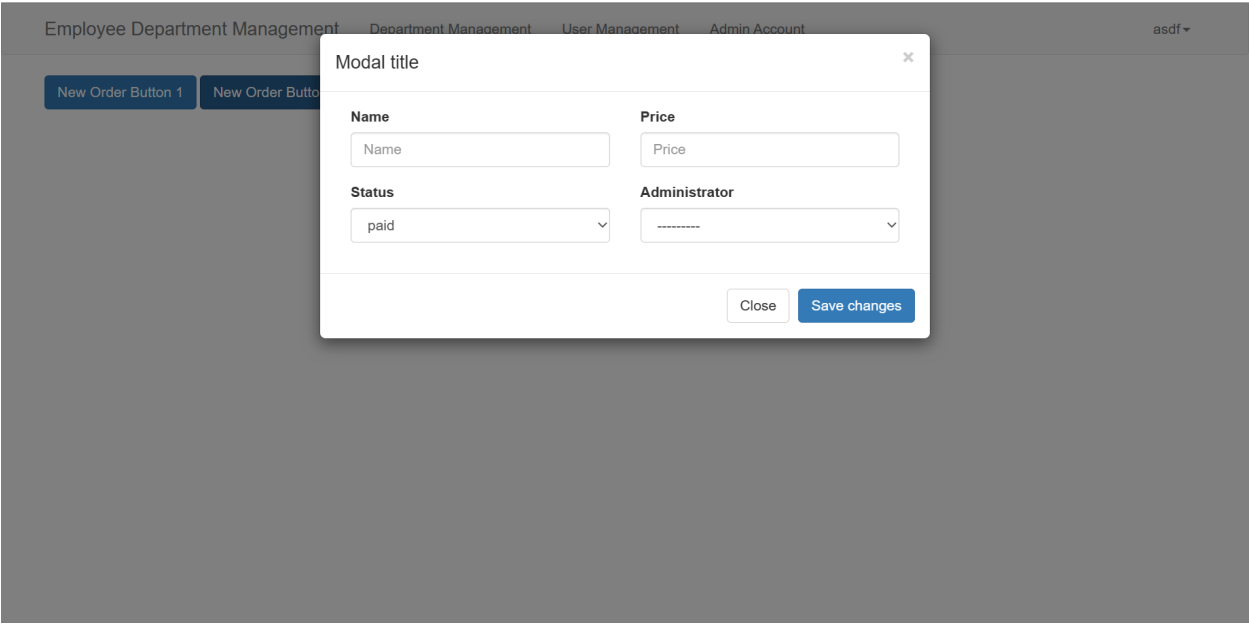
```
class OrderModelForm(BootstrapModelForm):

    class Meta:
        model = models.Order
        # fields = "__all__"
        exclude = ['oid']

def order_list(request):

    form = OrderModelForm()

    return render(request, 'order_list.html', {'form': form})
```



```
function bindBtnSaveEvent() {
    $("#btnSave").click(function () {
        $(".error_message").empty();

        $.ajax({
            url: "/order/add/",
            type: "post",
            data: $("#formAdd").serialize(),
            dataType: "JSON",
            success: function (res) {
                if(res.status){
                    alert('s')
                }else{
                    $.each(res.error, function (name, errorList) {
                        $("#id_"+name).next().text(errorList[0])
                    })
                }
            }
        })
    });
}
```

But we can not get oid

generate oid

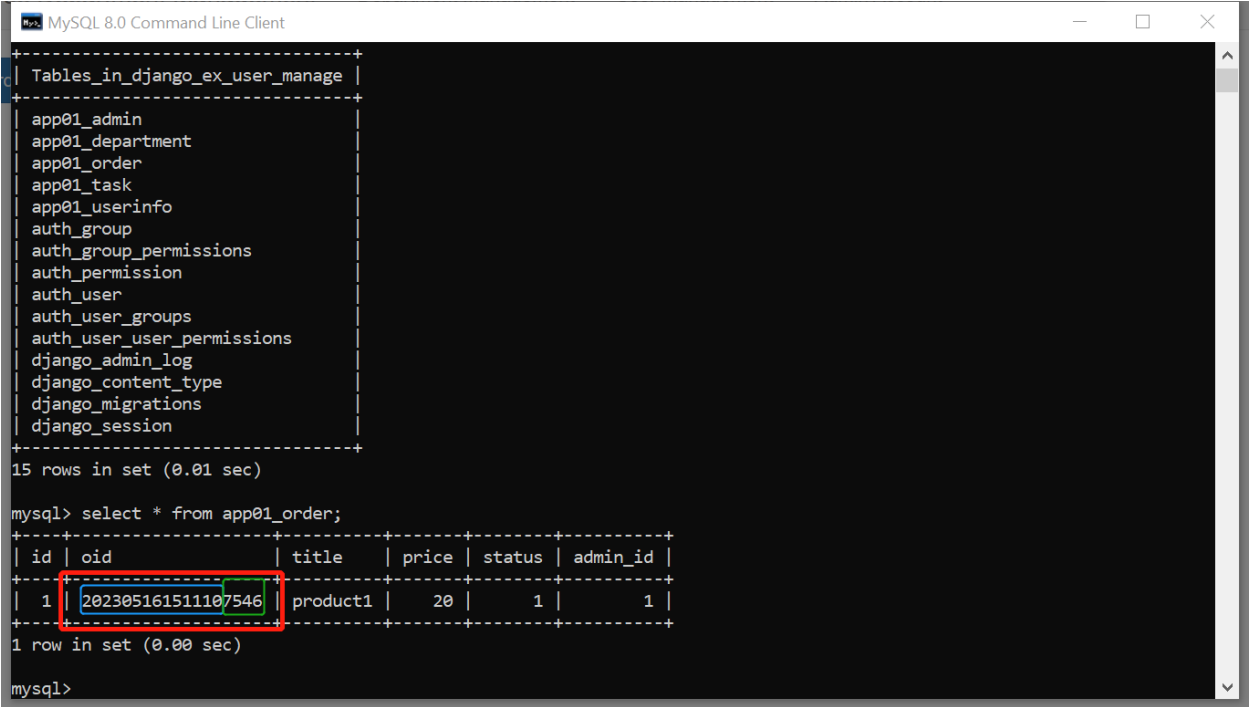
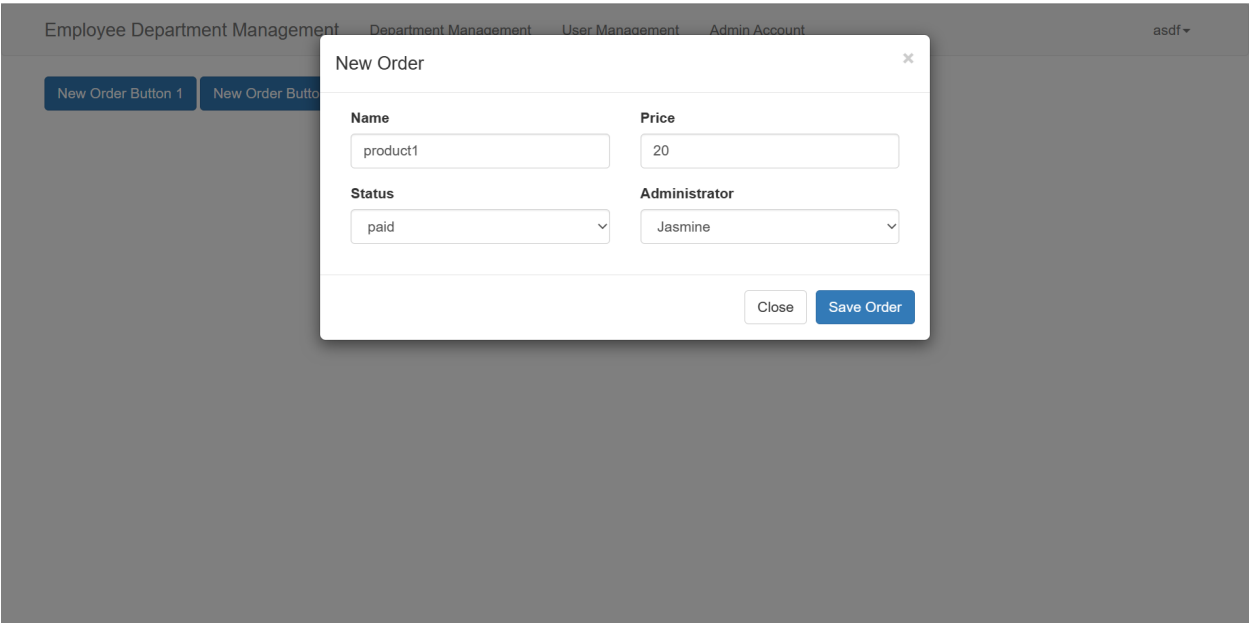
```
import random
from datetime import datetime

@csrf_exempt
def order_add(request):

    form = OrderModelForm(data=request.POST)
    if form.is_valid():
        form.instance.oid = datetime.now().strftime("%Y%m%d%H%M%S") + str(random.randint(1000, 9999))

        form.save()
        return JsonResponse({'status': True})

    return JsonResponse({'status': False, 'error': form.errors})
```



fix admin_id

use current login admin

```
import random
from datetime import datetime

@csrf_exempt
def order_add(request):

    form = OrderModelForm(data=request.POST)
    if form.is_valid():
        # add oid
        form.instance.oid = datetime.now().strftime("%Y%m%d%H%M%S") + str(random.randint(1000, 9999))

        # get current account id
        form.instance.admin_id = request.session['info']['id']

        form.save()
        return JsonResponse({'status': True})

    return JsonResponse({'status': False, 'error': form.errors})
```

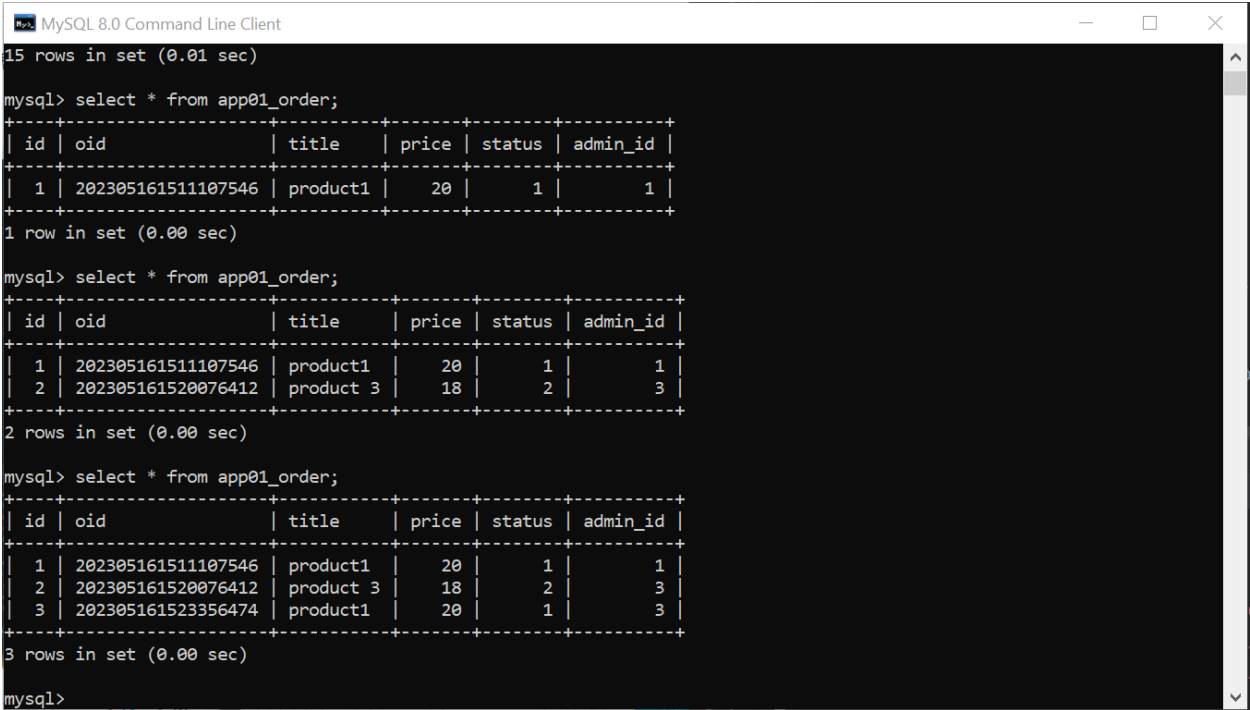
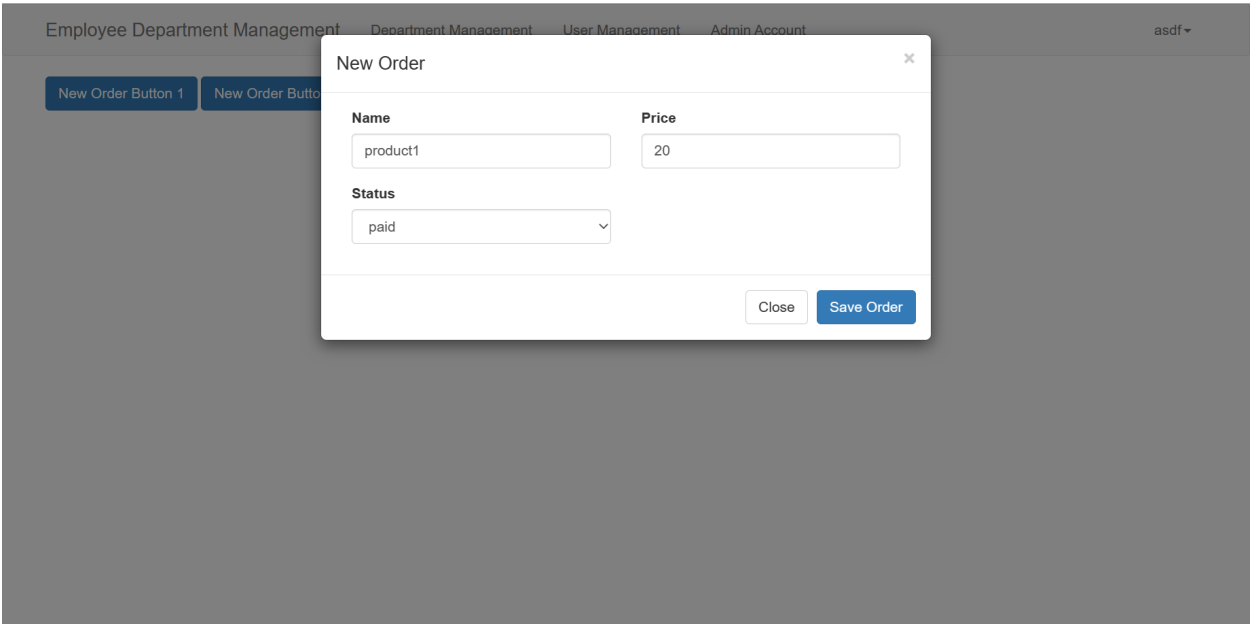
get admin id in session

```
class OrderModelForm(BootstrapModelForm):

    class Meta:
        model = models.Order
```

```
# fields = "__all__"
exclude = ['oid', 'admin']
```

Please pay attention, when doing exclude, the col name may be different from col name showing in databases (foreign key), should use value name when create tables.



delete all record when successfully add data

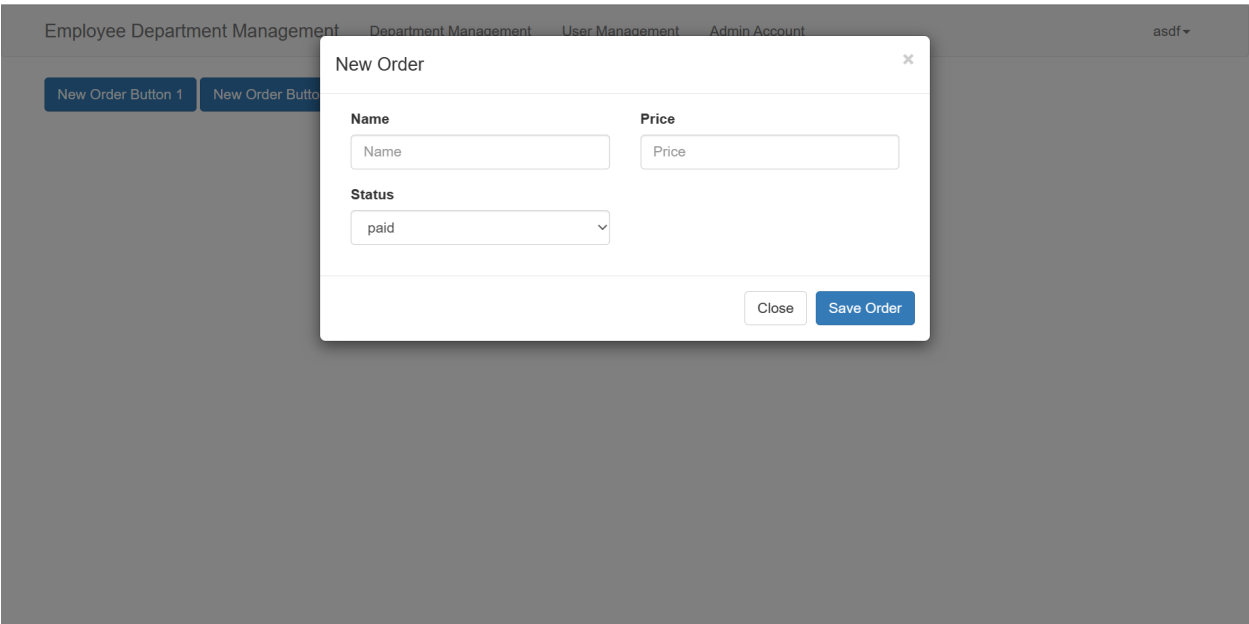
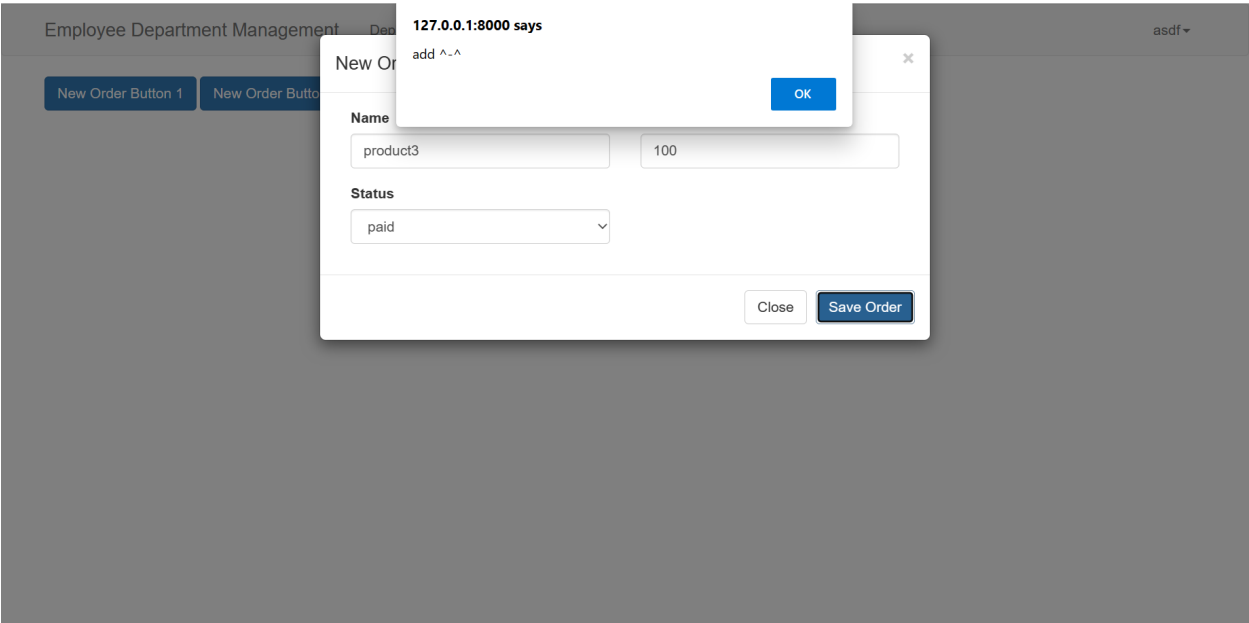
💡 `$("#formAdd")` → jquery object, `$("#formAdd")[0]` DOM object

💡 `$("#formAdd")[0].reset();`

```
function bindBtnSaveEvent() {
    $("#btnSave").click(function () {
        $(".error_message").empty();

        $.ajax({
            url: "/order/add/",
            type: "post",
            data: $("#formAdd").serialize(),
            dataType: "JSON",
            success: function (res) {
                if(res.status){
                    alert('add ^-^');
                }
            }
        });
    });
}
```

```
        $("#formAdd")[0].reset();
    }else{
        $.each(res.error, function (name, errorList) {
            $("#id_"+name).next().text(errorList[0])
        })
    }
}
});
}
```



close modal

```
💡 $('#myModal').modal('hide');
```

```
function bindBtnSaveEvent() {
    $("#btnSave").click(function (){
        $(".error_message").empty();

        $.ajax({
            url: "/order/add/",
            type: "post",
            data: $("#formAdd").serialize(),
            dataType: "JSON",
            success: function (res) {
                if(res.status){
                    //alert('add ^.^');
                    $("#formAdd")[0].reset();

                    $('#myModal').modal('hide');
                }else{
                    $.each(res.error, function (name, errorList) {
```



```

    }
    });
  })
}

$( "#id_"+name ).next().text( errorList[0] )
})
}
}

```

5. show orders

```
{% extends 'layout.html' %}

{% block content %}

<div class="container">
  <div>
    <input data-toggle="modal" data-target="#myModal" type="button" value="New Order Button 1"
      class="btn btn-primary">
    <input id="btnAdd" type="button" value="New Order Button 2" class="btn btn-primary">
  </div>
  <!-- Modal -->
  <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span></button>
          <h4 class="modal-title" id="myModalLabel">New Order</h4>
        </div>
        <div class="modal-body">
          <form id="formAdd">
            <div class="clearfix">
              {% for field in form %}
                <div class="col-xs-6">
                  <div class="form-group" style="position: relative">
                    <label>{{ field.label }}</label>
                    {{ field }}
                    <span class="error-message" style="color: red; position: absolute"></span>
                  </div>
                </div>
              {% endfor %}
            </div>
          </form>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
          <button id="btnSave" type="button" class="btn btn-primary">Save Order</button>
        </div>
      </div>
    </div>
  </div>

<div class="bs-example" data-example-id="panel-without-body-with-table">
  <div class="panel panel-default">
    <!-- Default panel contents -->
    <div class="panel-heading">Administrator List</div>

    <!-- Table -->
    <table class="table">
      <thead>
        <tr>
          <th>ID</th>
          <th>Order ID</th>
          <th>Product</th>
          <th>Price</th>
          <th>Status</th>
          <th>Administrator</th>
          <th>Operations</th>
        </tr>
      </thead>
      <tbody>
        {% for obj in queryset %}
          <tr>
            <th scope="row">{{ obj.id }}</th>
            <td>{{ obj.oid }}</td>
            <td>{{ obj.title }}</td>
            <td>{{ obj.price }}</td>
            <td>{{ obj.get_status_display }}</td>
            <td>{{ obj.admin }}</td>
            <td>
              <a class="btn btn-primary btn-sm" href="/admin/{{ obj.id }}/edit/">Edit</a>
              <a class="btn btn-danger btn-sm" href="/admin/{{ obj.id }}/delete/">Delete</a>
            </td>
          </tr>
        </tbody>
      </table>
    </div>
  </div>

</div>
</div>
</div>
```

```

                </tr>
                {% endfor %}

            </tbody>
        </table>
    </div>
</div>
<nav aria-label="Page navigation">
    <ul class="pagination">
        {{ page_string }}
    </ul>
</nav>

</div>

{% endblock %}
{% block js %}
<script type="text/javascript">
    $(function () {
        bindBtnAddEvent();
        bindBtnSaveEvent();
    })

    function bindBtnAddEvent() {
        $('#btnAdd').click(function () {
            $('#myModal').modal('show');
        });
    }

    function bindBtnSaveEvent() {
        $("#btnSave").click(function () {
            $(".error_message").empty();

            $.ajax({
                url: "/order/add/",
                type: "post",
                data: $("#formAdd").serialize(),
                dataType: "JSON",
                success: function (res) {
                    if(res.status){
                        //alert('add ^-^');
                        $("#formAdd")[0].reset();

                        $('#myModal').modal('hide');
                        // page reload
                        location.reload();
                    }else{
                        $.each(res.error, function (name, errorList) {
                            $("#id_"+name).next().text(errorList[0])
                        })
                    }
                }
            })
        });
    }

</script>

{% endblock %}

```

```

def order_list(request):
    queryset = models.Order.objects.all().order_by('-id')

    form = OrderModelForm()
    page_object = Pagination(request, queryset, page_size=5)

    context = {
        "form": form,
        "queryset": page_object.page_queryset,
        "page_string": page_object.html()
    }

    return render(request, 'order_list.html', context)

```

6. delete orders

alert

```

<!-- Modal -->
<div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel">
    <div class="modal-dialog" role="document">

```

```

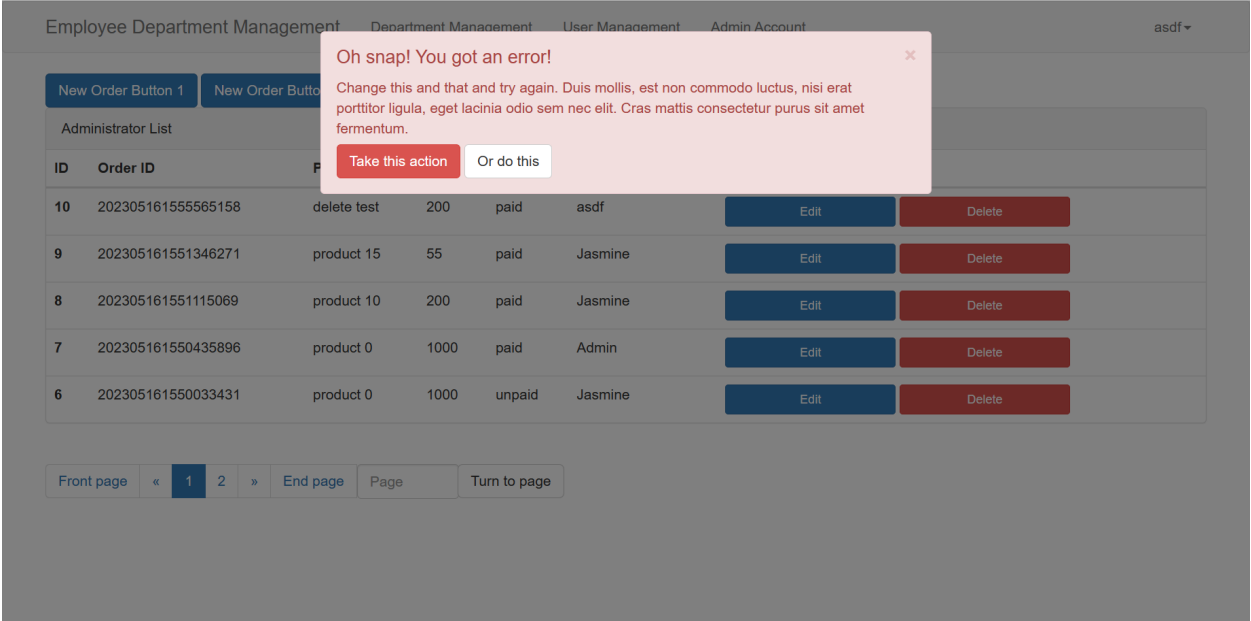
        <div class="alert alert-danger alert-dismissible fade in" role="alert">
            <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span
                aria-hidden="true">x</span></button>
            <h4>Oh snap! You got an error!</h4>
            <p>Change this and that and try again. Duis mollis, est non commodo luctus, nisi erat porttitor
                ligula, eget lacinia odio sem nec elit. Cras mattis consectetur purus sit amet fermentum.</p>
            <p>
                <button type="button" class="btn btn-danger">Take this action</button>
                <button type="button" class="btn btn-default">Or do this</button>
            </p>
        </div>
    </div>
</div>

```

```

function bindBtnDeleteEvent() {
    $(".btn-delete").click(function () {
        $("#deleteModal").modal('show');
    })
}

```



global

```
var DELETE_ID;
```

```

function bindBtnConfirmDelete() {
    $('#btnConfirmDelete').click(function () {
        $.ajax({
            url: "/order/delete/",
            type: "GET",
            data: {
                uid: DELETE_ID
            },
            dataType: "JSON",
            success: function (res) {
                if(res.status){
                    alert('delete ^-^');
                    $("#deleteModal").modal('hide');

                    // reset delete id
                    DELETE_ID = 0;

                }else{
                    alert(res.error);
                }
            }
        })
    });
}

```

delete record shown in the table



<tr uid="{{ obj.id }}">

add uid to each rows

uid is a custom variable, for finding the id



\$("tr[uid="" +DELETE_ID+ "']").remove();

```
function bindBtnConfirmDelete() {
    $('#btnConfirmDelete').click(function () {
        $.ajax({
            url: "/order/delete/",
            type: "GET",
            data: {
                uid: DELETE_ID
            },
            dataType: "JSON",
            success: function (res) {
                if(res.status){
                    alert('delete ^-^');
                    $('#deleteModal').modal('hide');

                    $("tr[uid="" +DELETE_ID+ "']").remove();

                    // reset delete id
                    DELETE_ID = 0;

                }else{
                    alert(res.error);
                }
            }
        })
    });
}
```

show table

Employee Department Management

Department Management

User Management

Admin Account

asdf

New Order Button 1

New Order Button 2

Administrator List

ID	Order ID	Product	Price	Status	Administrator	Operations
10	202305161555565158	delete test	200	paid	asdf	<div>Edit</div> <div>Delete</div>
9	202305161551346271	product 15	55	paid	Jasmine	<div>Edit</div> <div>Delete</div>
8	202305161551115069	product 10	200	paid	Jasmine	<div>Edit</div> <div>Delete</div>
7	202305161550435896	product 0	1000	paid	Admin	<div>Edit</div> <div>Delete</div>
6	202305161550033431	product 0	1000	unpaid	Jasmine	<div>Edit</div> <div>Delete</div>

Front page

«

1

2

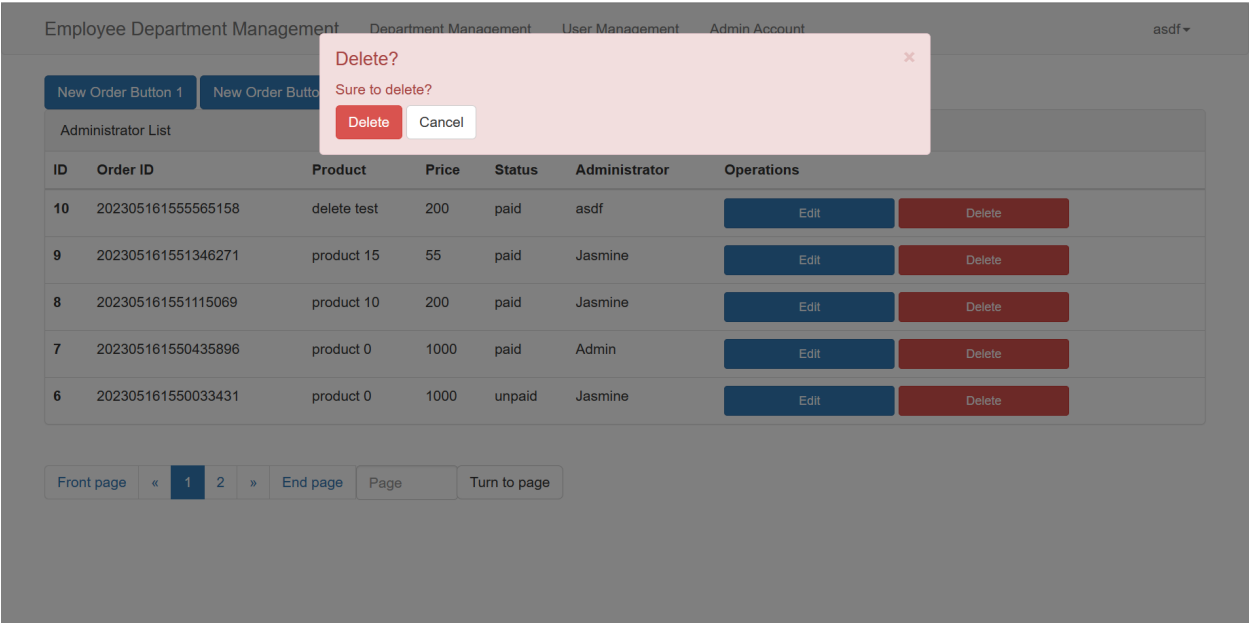
»

End page

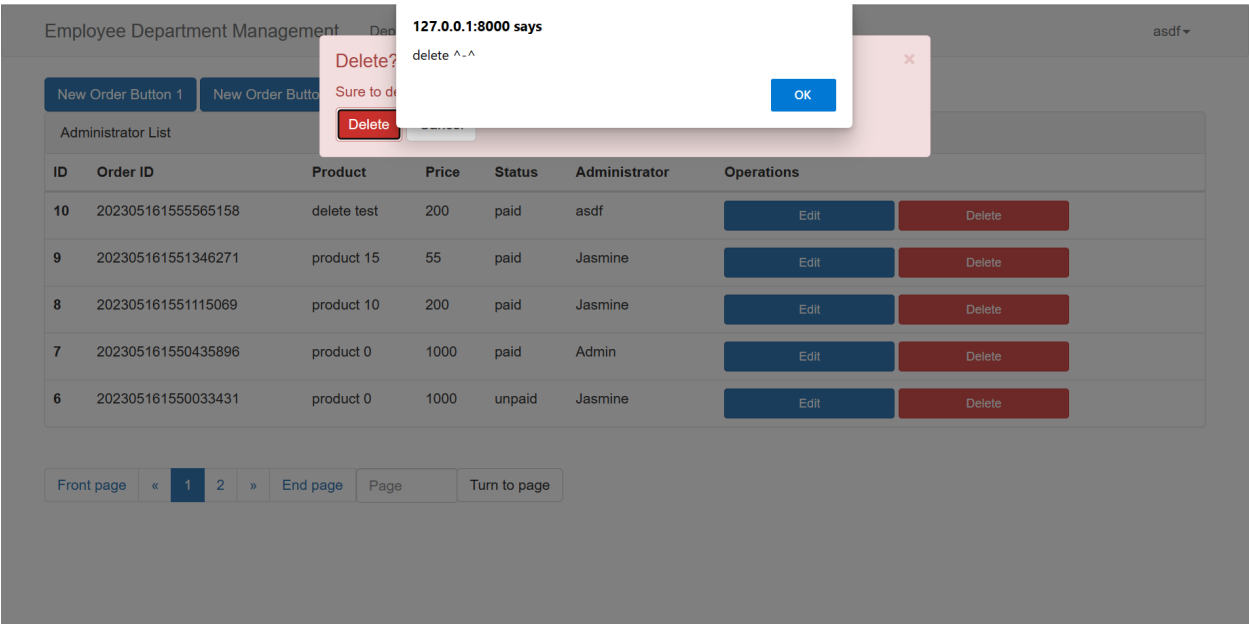
Page

Turn to page

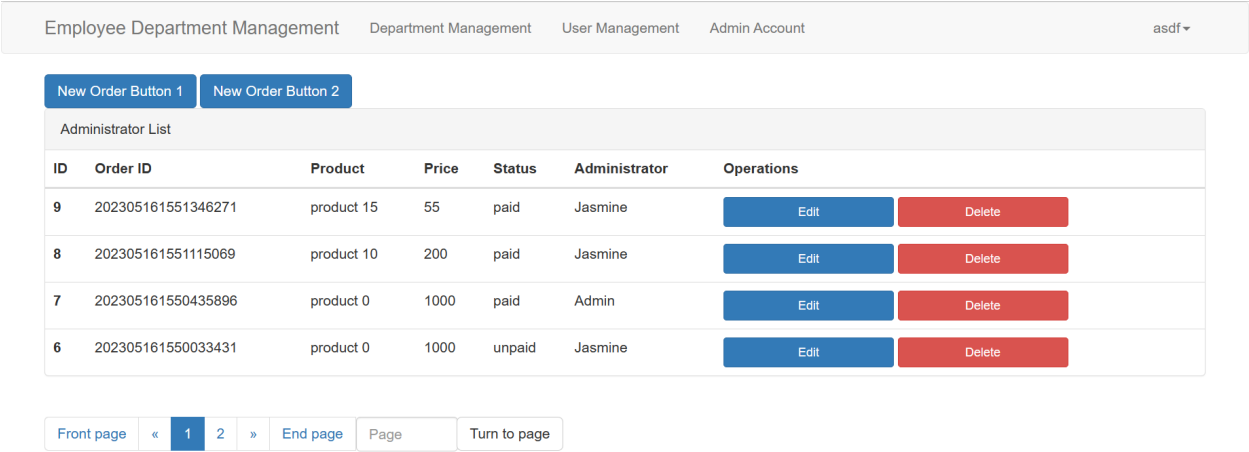
click delete



click delete, delete in database



click OK, delete in pages



But this method will not reload pages, thus, only original data in this page will show here, even after delete, other data will not enter this page.

Thus, we still need to reload pages.

→ A more simple way is to reload.

Employee Department Management

Department ManagementUser ManagementAdmin Account

asdf

New Order Button 1New Order Button 2

Administrator List

ID	Order ID	Product	Price	Status	Administrator	Operations
11	202305161636534571	new delete test	100	paid	asdf	<div>EditDelete</div>
9	202305161551346271	product 15	55	paid	Jasmine	<div>EditDelete</div>
8	202305161551115069	product 10	200	paid	Jasmine	<div>EditDelete</div>
7	202305161550435896	product 0	1000	paid	Admin	<div>EditDelete</div>
6	202305161550033431	product 0	1000	unpaid	Jasmine	<div>EditDelete</div>

Front page«12»End page

Page

Turn to page

Employee Department Management

Department ManagementUser ManagementAdmin Account

asdf

New Order Button 1New Order Button 2

Administrator List

ID	Order ID	Product	Price	Status	Administrator	Operations
11	202305161636534571	new delete test	100	paid	asdf	<div>EditDelete</div>
9	202305161551346271	product 15	55	paid	Jasmine	<div>EditDelete</div>
8	202305161551115069	product 10	200	paid	Jasmine	<div>EditDelete</div>
7	202305161550435896	product 0	1000	paid	Admin	<div>EditDelete</div>
6	202305161550033431	product 0	1000	unpaid	Jasmine	<div>EditDelete</div>

Front page«12»End page

Page

Turn to page

Delete?

Sure to delete?

DeleteCancel

Employee Department Management

Department ManagementUser ManagementAdmin Account

asdf

New Order Button 1New Order Button 2

Administrator List

ID	Order ID	Product	Price	Status	Administrator	Operations
11	202305161636534571	new delete test	100	paid	asdf	<div>EditDelete</div>
9	202305161551346271	product 15	55	paid	Jasmine	<div>EditDelete</div>
8	202305161551115069	product 10	200	paid	Jasmine	<div>EditDelete</div>
7	202305161550435896	product 0	1000	paid	Admin	<div>EditDelete</div>
6	202305161550033431	product 0	1000	unpaid	Jasmine	<div>EditDelete</div>

Front page«12»End page

Page

Turn to page

127.0.0.1:8000 says

delete ^-^

Delete?

Sure to delete?

DeleteCancel

OK

Employee Department Management

Department Management

User Management

Admin Account

asdf

New Order Button 1

New Order Button 2

Administrator List

ID	Order ID	Product	Price	Status	Administrator	Operations
9	202305161551346271	product 15	55	paid	Jasmine	<div>Edit</div> <div>Delete</div>
8	202305161551115069	product 10	200	paid	Jasmine	<div>Edit</div> <div>Delete</div>
7	202305161550435896	product 0	1000	paid	Admin	<div>Edit</div> <div>Delete</div>
6	202305161550033431	product 0	1000	unpaid	Jasmine	<div>Edit</div> <div>Delete</div>
5	202305161549468382	product 1	15	paid	Admin	<div>Edit</div> <div>Delete</div>

Front page

«

1

2

»

End page

Page

Turn to page

7. edit order

reuse add order modal

get data from data set, method 1:

```
uid = request.GET.get("uid")
row_object = models.Order.objects.filter(id=uid).first()
if not row_object:
    return JsonResponse({"status": False, 'error': "数据不存在。"})

# 从数据库中获取到一个对象 row_object
result = {
    "status": True,
    "data": {
        "title": row_object.title,
        "price": row_object.price,
        "status": row_object.status,
    }
}
return JsonResponse({"status": True, 'data': result})
```

method 2:

```
# 对象, 当前行的所有数据。
row_object = models.Order.objects.filter(id=uid).first()
row_object.id
row_object.title

# 字典, {"id":1,"title":"xx"}
row_dict = models.Order.objects.filter(id=uid).values("id","title").first()
```

支持 对象 字典 元组

想要去数据库中获取数据时：对象/字典

```
# 对象，当前行的所有数据。
row_object = models.Order.objects.filter(id=uid).first()
row_object.id
row_object.title

# 字典，{"id":1,"title":"xx"}
row_dict = models.Order.objects.filter(id=uid).values("id","title").first()
```

```
# queryset = [obj,obj,obj,]
queryset = models.Order.objects.all()

# queryset = [ {'id':1,'title':"xx"},{'id':2,'title':"xx"}, ]
queryset = models.Order.objects.all().values("id","title")

# queryset = [ (1,"xx"),(2,"xxx"), ]
queryset = models.Order.objects.all().values_list("id","title")
```

python

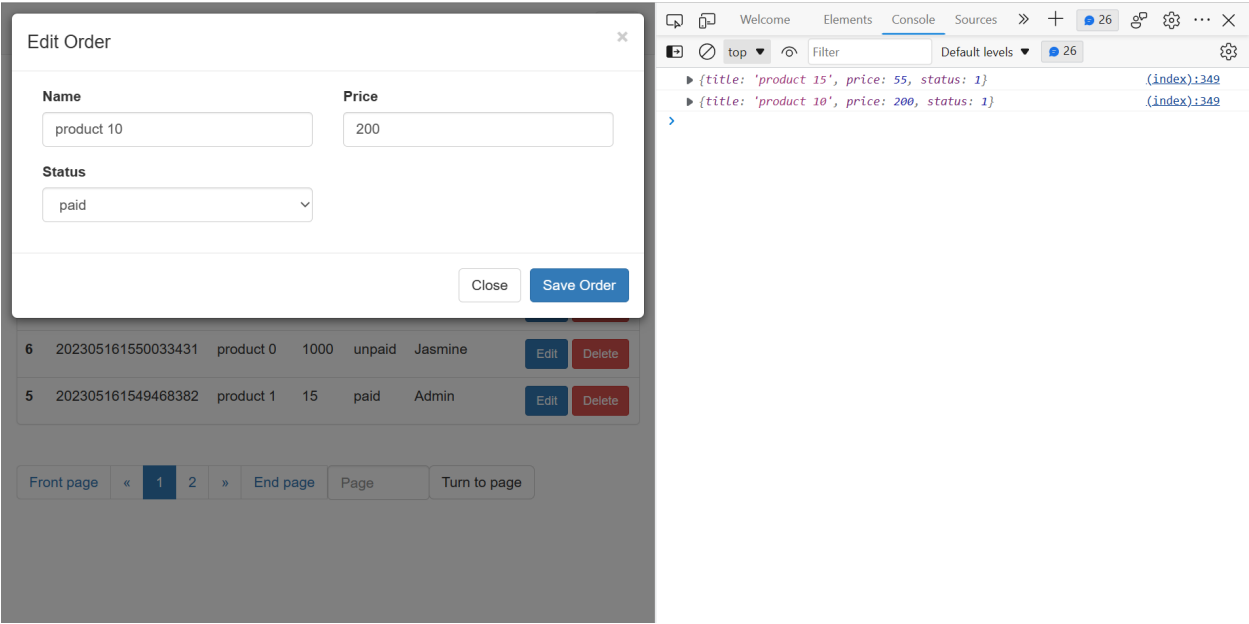
```
def order_detail(request):
    uid = request.GET.get('uid')
    row_dict = models.Order.objects.filter(id=uid).values('title', 'price', 'status').first()
    if not row_dict:
        JsonResponse({'status': False, 'error': 'No such order'})

    return JsonResponse({"status": True, 'data': row_dict})
```

```
function bindBtnEditEvent() {
    $(".btn-edit").click(function () {
        var uid = $(this).attr("uid");
        $("#myModallabel").text('Edit Order');

        // get info show in input
        $.ajax({
            url: '/order/detail/',
            type: "get",
            data: {
                uid: uid
            },
            dataType: "JSON",
            success: function (res) {
                if(res.status){
                    console.log(res.data);
                    // add previous data to modal before open modal
                    $.each(res.data, function (name, value) {

                        $("#id_"+name).val(value);
                    })
                    $("#myModal").modal('show');
                }else{
                    alert(res.error);
                }
            }
        })
    });
}
```

combine add & edit

```
function bindBtnAddEvent() {
    $('#btnAdd').click(function () {
        // reset edit id to undefined
        EDIT_ID = undefined;
        // because edit & add used the same modal
        // delete all data when add order
        // rename modal title
        $("#modalAdd")[0].reset();
        $("#myModalLabel").text('New Order');
        $('#myModal').modal('show');
    });
}
```

```
function bindBtnSaveEvent() {
    $("#btnSave").click(function () {
        $(".error_message").empty();
        if (EDIT_ID) {
            doEdit();
        } else {
            doAdd();
        }
    });
}

function doAdd() {
    // add request
    $.ajax({
        url: "/order/add/",
        type: "post",
        data: $("#formAdd").serialize(),
        dataType: "JSON",
        success: function (res) {
            if (res.status) {
                //alert('add ^-^');
                $("#formAdd")[0].reset();

                $('#myModal').modal('hide');
                // page reload
                location.reload();
            } else {
                $.each(res.error, function (name, errorList) {
                    $("#id_" + name).next().text(errorList[0])
                })
            }
        }
    })
}

function doEdit() {
    // add request
    $.ajax({
        url: "/order/edit/" + "?uid=" + EDIT_ID,
        type: "post",
        data: $("#formAdd").serialize(),
        dataType: "JSON",
        success: function (res) {
            if (res.status) {
                //alert('add ^-^');
            }
        }
    })
}
```

```

        $("#formAdd")[0].reset();

        $('#myModal').modal('hide');
        // page reload
        location.reload();
    } else {
        if (res.tips) {
            alert(res.tips);
        } else {
            $.each(res.error, function (name, errorList) {
                $("#id_" + name).next().text(errorList[0])
            })
        }
    }
}
})
}

```

```

function bindBtnEditEvent() {
    $(".btn-edit").click(function () {
        // clean input
        $("#formAdd")[0].reset();
        var uid = $(this).attr("uid");
        $("#myModalLabel").text('Edit Order');

        // get uid
        EDIT_ID = uid;

        // get info show in input
        $.ajax({
            url: '/order/detail/',
            type: "get",
            data: {
                uid: uid
            },
            dataType: "JSON",
            success: function (res) {
                if (res.status) {
                    // add previous data to modal before open modal
                    $.each(res.data, function (name, value) {

                        $("#id_" + name).val(value);
                    })
                    $("#myModal").modal('show');
                } else {
                    alert(res.error);
                }
            }
        })
    });
}

```