



Deng Admin 개발 기능 (상세)

- 게시판 카테고리 관리

1 페이지를 로드할 때, ajax 를 통해 카테고리 정보를 뿌려준다.

1-1
상태에 따라  는 현재 서비스 중을 나타내고,
 는 서비스 중지 상태를 나타낸다.

```
temp[2] = (categories[i].status == 200) ?  
    '<span data-feather="thumbs-up" style="color:green"></span>' :  
    '<span data-feather="thumbs-down" style="color:red"></span>';
```

2 ajax로 해당 카테고리의 상태를 변경하고 테이블을 다시 세팅한다.

```
function changeStatus(seq, status) {  
    $.ajax({  
        url: 'AdminController',  
        data: { command: 'changeState', seq: seq, status: status },  
        method: 'POST',  
        success: function (data) {  
            var result = JSON.parse(data);  
  
            if(result){  
                if($('#alertDiv').hasClass('alert-danger')){  
                    $('#alertDiv').removeClass('alert-danger');  
                }  
                $('#alertDiv').addClass('alert-success');  
                $('#alertDiv').html('상태가 변경되었습니다.');            } else {  
                if($('#alertDiv').hasClass('alert-success')){  
                    $('#alertDiv').removeClass('alert-success');  
                }  
                $('#alertDiv').addClass('alert-danger');  
                $('#alertDiv').html('상태가 변경할 수 없습니다.');            }  
            $('#alertDiv').css('display', 'block');  
  
            $('#category-title').val('');  
            $('#category-discription').val('');  
            $('#myTable').DataTable().clear().draw();  
            drawTable();  
        }  
    })  
}
```

Deng Admin 개발 기능 (상세)

- 게시판 카테고리 관리

The screenshot shows the 'Category' management interface in the Deng Admin system. On the left is a sidebar with navigation links. The main area displays a table of existing categories. To the right, a form for adding a new category is highlighted with a red box and a red circle containing the number 3. The form includes a 'Title' text input, a 'Status' section with 'Active' (selected) and 'Non Active' radio buttons, a 'Description' text area, and a '카테고리 추가하기' (Add Category) button.

3 해당 필드를 입력하고 추가하기 버튼을 누르면, submit을 javascript 부분으로 넘긴다.

```
<div class="col-md-4">
  <form id="categoryForm" onsubmit="return false">
    <input type="hidden" name="command" value="addCategory">
    <div class="form-group">
      <label for="category-title">Title</label>
      <input type="text" name="title" class="form-control" id="category-title" placeholder="Category Title">
    </div>

    <label for="category-status">Status</label>
    <div class="form-group" style="text-align:center;">
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="status" id="status1" value="200" checked>
        <label class="form-check-label" for="status1"> Active </label>
      </div>
      <div class="form-check form-check-inline">
        <input class="form-check-input" type="radio" name="status" id="status2" value="404">
        <label class="form-check-label" for="status2"> Non Active </label>
      </div>
    </div>
    <div class="form-group">
      <label for="category-description">Description</label>
      <textarea class="form-control" name="description" id="category-description" rows="3"></textarea>
    </div>
    <div class="form-group">
      <button class="btn btn-outline-primary" id="addCategory" style="width:100%">카테고리 추가하기</button>
    </div>
  </form>
</div>
```

3 -1 form 부분을 serialize() 하여 값들을 ajax로 전송한다.

```
$('#addCategory').on('click', function () {
  var FormData = $('#categoryForm').serialize();
  $.ajax({
    url : 'AdminController',
    data : FormData,
    method : 'POST',
    success : function (data) {
      var result = JSON.parse(data);

      if(result){
        if($('#alertDiv').hasClass('alert-danger')){
          $('#alertDiv').removeClass('alert-danger');
        }
        $('#alertDiv').addClass('alert-success');
        $('#alertDiv').html('상태가 변경되었습니다.');
```