

HƯỚNG DẪN AJAX VỚI RESTFUL API TRONG SPRING BOOT

1. THÊM THƯ VIỆN AJAX: MỞ TRANG LAYOUT WEB (admin.jsp, web.jsp) thêm 02 lệnh này vào thẻ <head> phiên bản ajax tùy thuộc vào template bootstrap đang sử dụng

```
<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
<script>var contextPath = "${pageContext.request.contextPath}"</script>
```

Ví dụ: trang admin.jsp sử dụng bootstrap 5

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ include file="/common/taglib.jsp"%>
<!DOCTYPE html>
<html>
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
    rel="stylesheet"
    integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpUuCOmLASjC"
    crossorigin="anonymous">
<link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/font-awesome@6.0.0-beta2/css/all.min.css"
    integrity="sha512-
YWzhKL2whUzgiheMoBFwW8CKV4qphQAEuviLg9FAn5VJUDwKZZxkJNuGM4XkKwuk94WCrrwsLk8yWNGmY1EduT
A=="
    crossorigin="anonymous" referrerpolicy="no-referrer" />
<link rel="stylesheet" href="/css/mystyles.css" />
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/font-awesome@6.1.1/css/all.min.css" integrity="sha512-
KfkfwYDSLkILwQp6LFLN8zNdLGxu9YAA1QvWINKs4PhcELQSVqcyVLLD9aMhXd13uQjoXtEKNosOWaZqXgel0
g==" crossorigin="anonymous" referrerpolicy="no-referrer" />
    <link href="https://unpkg.com/boxicons@2.0.9/css/boxicons.min.css"
    rel="stylesheet" />
    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>
    <script>var contextPath = "${pageContext.request.contextPath}"</script>
<title>Index</title>
</head>
<body>

    <header class="row">
        <div class="col">
            <%@include file="/common/admin/header.jsp"%>
        </div>
    </header>
    <main class="container-fluid">
        <!-- <nav class="row">
            <div class="col">
                <div></div>
            </div>
        </nav> -->
        <div class="row mt-4">
            <div class="col mt-5 mb-5">
```

```

<sitemesh:write property='body'></sitemesh:write>

    </div>
</div>

</main>
<footer class="row">
    <div class="col">
        <%@include file="/common/admin/footer.jsp" %>
    </div>
</footer>

<script

    src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min
.js"
    integrity="sha384-
Mrcw6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
    crossorigin="anonymous"></script>

</body>
</html>

```

2. Controller API

```

package vn.iotstar.controllers.api;

import java.util.Optional;
import java.util.UUID;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import vn.iotstar.entity.Category;
import vn.iotstar.model.Response;
import vn.iotstar.service.ICategoryService;
import vn.iotstar.service.IStorageService;

@RestController
@RequestMapping(path = "/api/category")
public class CategoryApiController {

    @Autowired
    private ICategoryService categoryService;

```

```

@Autowired
IStorageService storageService;

@GetMapping
public ResponseEntity<?> getAllCategory() {
    return ResponseEntity.ok().body(categoryService.findAll());
    //return new ResponseEntity<Response>(new Response(true, "Thành
công",categoryService.findAll()), HttpStatus.OK);
}

@PostMapping(path = "/getCategory")
public ResponseEntity<?> getCategory(@Validated @RequestParam("id") Long id) {
    Optional<Category> category = categoryService.findById(id);

    if (category.isPresent()) {
        // return ResponseEntity.ok().body(category.get());
        return new ResponseEntity<Response>(new Response(true, "Thành
công", category.get()), HttpStatus.OK);
    } else {
        // return ResponseEntity.notFound().build();
        return new ResponseEntity<Response>(new Response(false, "Thất
bại", null), HttpStatus.NOT_FOUND);
    }
}

@PostMapping(path = "/addCategory")
public ResponseEntity<?> addCategory(@Validated @RequestParam("categoryName")
String categoryName,
    @Validated @RequestParam("icon") MultipartFile icon) {
    Optional<Category> optCategory =
categoryService.findByName(categoryName);

    if (optCategory.isPresent()) {
        return
ResponseEntity.status(HttpStatus.BAD_REQUEST).body("Category đã tồn tại trong hệ
thống");
        // return new ResponseEntity<Response>(new Response(false, "Loại
sản phẩm này đã
        // tồn tại trong hệ thống", optCategory.get()),
HttpStatus.BAD_REQUEST);
    } else {
        Category category = new Category();
        // kiểm tra tồn tại file, lưu file
        if (!icon.isEmpty()) {
            UUID uuid = UUID.randomUUID();
            String uuString = uuid.toString();
            // lưu file vào trường Images
            category.setIcon(storageService.getSorageFilename(icon,
uuString));
            storageService.store(icon, category.getIcon());
        }

        category.setCategoryName(categoryName);

        categoryService.save(category);
        // return ResponseEntity.ok().body(category);
        return new ResponseEntity<Response>(new Response(true, "Thêm
Thành công", category), HttpStatus.OK);
    }
}

```

```

    @PutMapping(path = "/updateCategory")
    public ResponseEntity<?> updateCategory(@Validated @RequestParam("categoryId")
Long categoryId,
        @Validated @RequestParam("categoryName") String categoryName,
        @Validated @RequestParam("icon") MultipartFile icon) {
        Optional<Category> optCategory = categoryService.findById(categoryId);

        if (optCategory.isEmpty()) {
            return new ResponseEntity<Response>(new Response(false, "Không
tìm thấy Category", null),
                    HttpStatus.BAD_REQUEST);
        } else if (optCategory.isPresent()) {

            // kiểm tra tồn tại file, lưu file
            if (!icon.isEmpty()) {
                UUID uuid = UUID.randomUUID();
                String uuString = uuid.toString();
                // lưu file vào trường Images

                optCategory.get().setIcon(storageService.getSorageFilename(icon, uuString));
                storageService.store(icon, optCategory.get().getIcon());
            }

            optCategory.get().setCategoryName(categoryName);
            categoryService.save(optCategory.get());
            // return ResponseEntity.ok().body(category);
            return new ResponseEntity<Response>(new Response(true, "Cập nhật
Thành công", optCategory.get()),
                    HttpStatus.OK);
        }
        return null;
    }

    @DeleteMapping(path = "/deleteCategory")
    public ResponseEntity<?> deleteCategory(@Validated @RequestParam("categoryId")
Long categoryId) {
        Optional<Category> optCategory = categoryService.findById(categoryId);
        if (optCategory.isEmpty()) {
            return new ResponseEntity<Response>(new Response(false, "Không
tìm thấy Category", null),
                    HttpStatus.BAD_REQUEST);
        } else if (optCategory.isPresent()) {
            categoryService.delete(optCategory.get());
            return ResponseEntity.ok().body(optCategory.get());
            //return new ResponseEntity<Response>(new Response(true, "Xóa
Thành công", optCategory.get()), HttpStatus.OK);
        }

        return null;
    }
}

package vn.iotstar.controllers.api;

import java.sql.Timestamp;
import java.util.Date;

import java.util.Optional;
import java.util.UUID;

```

```

import org.springframework.beans.BeanUtils;
import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;

import org.springframework.validation.annotation.Validated;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import vn.iotstar.entity.Category;
import vn.iotstar.entity.Product;
import vn.iotstar.model.ProductModel;
import vn.iotstar.model.Response;

import vn.iotstar.service.ICategoryService;
import vn.iotstar.service.IProductService;
import vn.iotstar.service.IStorageService;

@RestController
@RequestMapping(path = "/api/product")
public class ProductApiController {

    @Autowired
    IProductService productService;
    @Autowired
    ICategoryService categoryService;

    @Autowired
    IStorageService storageService;

    @GetMapping
    public ResponseEntity<?> getAllProduct() {
        return new ResponseEntity<Response>(new Response(true, "Thành công",
productService.findAll()),
        HttpStatus.OK);
    }

    @PostMapping(path = "/addProduct")
    public ResponseEntity<?> saveOrUpdate(
        @Validated @RequestParam("productName") String productName,
        @RequestParam("imageFile") MultipartFile productImages,
        //@RequestParam("images") String images,
        @Validated @RequestParam("unitPrice") Double productPrice,
        @Validated @RequestParam("discount") Double promotionalPrice,
        @Validated @RequestParam("description") String
productDescription,
        @Validated @RequestParam("categoryId") Long categoryId,
        @Validated @RequestParam("quantity") Integer quantity,
        @Validated @RequestParam("status") Short status) {
        Optional<Product> optProduct =
productService.findByProductName(productName);

        if (optProduct.isPresent()) {

```

```

        // return ResponseEntity.status(HttpStatus.BAD_REQUEST).body("Sản
phẩm này đã
        // tồn tại trong hệ thống");
        return new ResponseEntity<Response>(
            new Response(false, "Sản phẩm này đã tồn tại trong
hệ thống", optProduct.get()),
            HttpStatus.BAD_REQUEST);
    } else {
        Product product = new Product();
        Timestamp timestamp = new Timestamp(new
Date(System.currentTimeMillis()).getTime());

        try {
            ProductModel proModel = new ProductModel();
            //copy từ Model sang Entity
            BeanUtils.copyProperties(proModel, product);
            //xử lý category liên quan product
            Category cateEntity = new Category();
            cateEntity.setCategoryId(proModel.getCategoryId());
            product.setCategory(cateEntity);
            //kiểm tra tồn tại file, lưu file
            if(!proModel.getImageFile().isEmpty()) {
                UUID uuid = UUID.randomUUID();
                String uuString = uuid.toString();
                //lưu file vào trường Images

                product.setImages(storageService.getSorageFilename(proModel.getImageFile(),
uuString));
                storageService.store(proModel.getImageFile(),
product.getImages());
            }
            product.setCreateDate(timestamp);

            productService.save(product);

            optProduct = productService.findByCreateDate(timestamp);

        } catch (Exception e) {
            e.printStackTrace();
        }
        // return ResponseEntity.ok().body(product);
        return new ResponseEntity<Response>(new Response(true, "Thành
công", optProduct.get()), HttpStatus.OK);
    }
}
}

```

3. Thêm trang jsp để hiện danh sách Category, thêm category, xóa category, ví dụ ajax.jsp

Hiển thị danh sách category bằng API và Ajax

```

<table class="table table-striped table-responsive">
  <thead class="thead-inverse">
    <tr>
      <th>Id</th>
      <th>icon</th>

```

```

                <th>Name</th>
                <th>Actions</th>
            </tr>
        </thead>
    </table>

<script type="text/javascript">
    /* Get category */
    $(document).ready(function() {
        $.getJSON(contextPath + '/api/category', function(json) {
            var tr=[];
            for (var i = 0; i < json.length; i++) {
                tr.push('<tr>');
                tr.push('<td>' + json[i].categoryId + '</td>');
                tr.push('<td>' + '</td>');
                tr.push('<td>' + json[i].categoryName + '</td>');
                tr.push('<td>' + '<a href="#" data-id="' +
json[i].categoryId+' " id="editcate" class="btn btn-outline-warning"><i class="fa fa-
edit"></i></a>'
                                + '<a href="#" data-id="' +
json[i].categoryId+' " id="categoryId" class="btn btn-outline-danger"><i class="fa fa-
trash"></i></a>');
                tr.push('</tr>');
            }
            $('table').append($(tr.join('')));
        });
    });
</script>

```

=====

Thêm category bằng API và Ajax sử dụng Modal bootstrap

```

        <p><button class="btn btn-success ml-auto"
onclick="showCreateNewCategoryModal()"><i class="fas fa-plus mr-2"></i>Thêm Category
Ajax</button></p>

<div class="modal" tabindex="-1" role="dialog" id="createCategoryModal">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <form id="addCategory" method="post" onsubmit="return false;"
enctype="multipart/form-data">
                <div class="modal-header">
                    <h5 class="modal-title">Add Category</h5>
                    <button type="button" class="close" data-dismiss="modal">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
                <div class="modal-body">
                    <div class="form-group">
                        <label for="new_categoryname">Category
Name</label>
                        <input type="text" class="form-control"
id="new_categoryname" name="categoryName">
                    </div>
                    <div class="form-group">
                        <label for="new_icon">Icon</label>

```

```

        <input type="file" class="form-control"
id="new_icon" name="icon">
    </div>
    <div class="form-group row">
        <div class="col text-center">
            <button type="submit" class="btn btn-
primary btn-block">Add</button>
        </div>
    </div>

    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary"
data-dismiss="modal">Đóng</button>
    </div>
</form>
</div>

```

```

</div>
<script type="text/javascript">

    /* Add category */
    $("form#addCategory").submit(function(e) {
        e.preventDefault();
        var formData = new FormData(this);

        $.ajax({
            url: contextPath + '/api/category/addCategory',
            type: 'POST',
            dataType: "json",
            data: formData,
            success: function (data) {
                location.reload();
            },
            cache: false,
            contentType: false,
            processData: false
        });
    });

    function showCreateNewCategoryModal(){
        $('#new_categoryname')[0].value = '';
        $('#createCategoryModal').modal('show');
    }

</script>

```

=====

Cập nhật Category bằng API và Ajax với Modal

```

<div class="modal" tabindex="-1" role="dialog" id="updateCategoryInfoModal">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title">Update Category</h5>
                <button type="button" class="close" data-dismiss="modal">
                    <span aria-hidden="true">&times;</span></button>
            </div>

```



```

        </div>
        <div class="modal-body">
            <div class="card">
                <div class="card-header">
                    <h5><i class="far fa-address-card mr-
1"></i>Category</h5>

                </div>
                <div class="card-body pb-0">
                    <p id="updateCategoryInfoModalId"></p>
                    <p id="updateCategoryInfoModalName"></p>
                    <p id="updateCategoryInfoModalIcon"></p>
                </div>
            </div>

            <div class="card mt-2">
                <div class="card-header">
                    <h5><i class="far fa-address-card mr-
1"></i>Update Category</h5>

                </div>
                <div class="card-body pb-0">
                    <form id="updateCategory" method="post"
onsubmit="return false;" enctype="multipart/form-data">
                        <div class="form-group">
                            <label
for="categoryName">Category Name</label>
                            <input type="text" class="form-
control" id="categoryName_up" name="categoryName">
                        </div>
                        <div class="form-group">
                            <label for="new_icon">Icon</label>
                            <input type="file" class="form-control"
id="icon_up" name="icon">
                        </div>
                        <input type="hidden" id="categoryId_up"
name="categoryId">
                        <div class="form-group row">
                            <div class="col text-center">
                                <button type="submit"
class="btn btn-primary btn-block" >Cập nhật</button>
                            </div>
                        </div>
                    </form>
                </div>
            </div>

            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
data-dismiss="modal">Đóng</button>
            </div>
        </div>
    </div>
</div>

```

```
<script type="text/javascript">
```

```

    /*      Edit */
    function showEditCategoryModal(categoryId, categoryName, icon, btn){

```

```

        /* $("input[name=categoryActive][value=" + active + "]).prop('checked',
true); */
        $('#updateCategoryInfoModalId')[0].innerText = "Category ID: " +
categoryId;
        $('#updateCategoryInfoModalName')[0].innerText = "Category Name: " +
categoryName;
        $('#updateCategoryInfoModalIcon')[0].innerText = 'Icon: ' + icon;
        /* $('#updateCategoryInfoModalActive')[0].innerText = 'Icon: ' + (active
== 'true' ? 'Đang cho phép bán' : 'Đã vô hiệu hóa'); */

        $('#categoryName_up')[0].value = categoryName;
        $('#categoryId_up')[0].value = categoryId;

        $('#updateCategoryInfoModal').modal('show');
    }

    $("form#updateCategory").submit(function(e) {
        e.preventDefault();
        var formData = new FormData(this);

        $.ajax({
            url: contextPath + '/api/category/updateCategory',
            type: 'PUT',
            dataType: "json",
            data: formData,
            success: function (data) {
                location.reload();
            },
            cache: false,
            contentType: false,
            processData: false
        });
    });
</script>

```

Xóa Category bằng API và Ajax

```

<a href="#" data-id="'+ json[i].categoryId+' " id="categoryId" class="btn btn-outline-
danger"><i class="fa fa-trash"></i></a>
<script type="text/javascript">

        $(document).delegate('#categoryId', 'click', function() {
            var id = $(this).data('id');
            if (confirm('Do you really want to delete record?')) {
                var parent = $(this).parent().parent();
                $.ajax({
                    type: "DELETE",
                    url: contextPath +
'/api/category/deleteCategory?categoryId=' + id,
                    dataType: "json",
                    data:id,
                    success: function() {
                        parent.fadeOut('slow', function() {
                            $(this).remove();
                        });
                        location.reload(true)
                    },
                    error: function() {

```

```
        });  
    }  
});  
});  
</script>
```

Bài tập thêm: Xây dựng chức năng CRUD bảng Products bằng Ajax