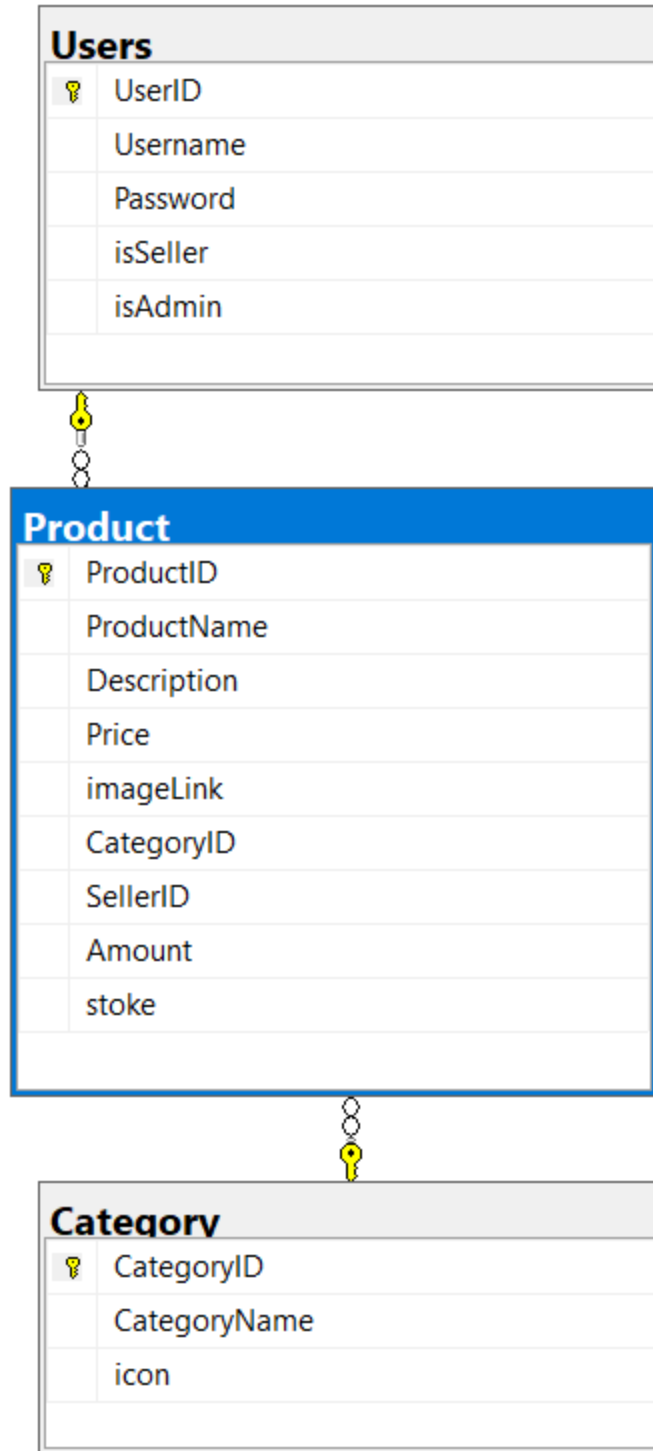


HƯỚNG DẪN RESTFULL API TRÊN SERVLET JPA VÀ SPRING BOOT

1. HƯỚNG DẪN CRUD CATEGORY TRÊN SERVLET JPA VỚI THƯ VIỆN JACKSON



Bước 1: thêm thư viện

```
<!-- json -->
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-core</artifactId>
  <version>2.6.3</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
  <version>2.6.3</version>
</dependency>
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-annotations</artifactId>
  <version>2.6.3</version>
</dependency>

<dependency>
  <groupId>org.codehaus.jackson</groupId>
  <artifactId>jackson-mapper-asl</artifactId>
  <version>1.9.13</version>
</dependency>
<dependency>
  <groupId>org.codehaus.jackson</groupId>
  <artifactId>jackson-core-asl</artifactId>
  <version>1.9.13</version>
</dependency>
<!--end json -->
```

Bước 2: Xây dựng Entity CATEGORY và PRODUCT

```
package vn.iotstar.entity;

import java.io.Serializable;
import javax.persistence.*;

import com.fasterxml.jackson.annotation.JsonManagedReference;

import java.util.List;

@Entity
@Table(name="Category")
@NamedQuery(name="Category.findAll", query="SELECT c FROM Category c")
public class Category implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int CategoryID;
    @Column(columnDefinition = "nvarchar(200)")
    private String CategoryName;
    @Column(columnDefinition = "nvarchar(MAX)")
    private String icon;
```

```

        @JsonManagedReference
        @OneToMany(mappedBy="category", fetch=FetchType.EAGER)
        private List<Product> products;

//tạo getters and setter và constructor

}
=====
package vn.iotstar.entity;

import java.io.Serializable;
import javax.persistence.*;

import com.fasterxml.jackson.annotation.JsonIgnore;

@Entity
@Table(name="Product")
@NamedQuery(name="Product.findAll", query="SELECT p FROM Product p")
public class Product implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int ProductID;
    private int stoke;
    private int Amount;
    @Column(columnDefinition = "nvarchar(MAX)")
    private String Description;
    @Column(columnDefinition = "nvarchar(255)")
    private String ProductName;
    private String imageLink;
    private int SellerID;
    private int Price;

    @JsonIgnore
    @ManyToOne
    @JoinColumn(name="CategoryID")
    private Category category;

//tạo getters and setter và constructor

}

```

Bước 3: xây dựng tầng DAO

```

package vn.iotstar.Dao;

import java.util.List;

import vn.iotstar.entity.Category;

public interface ICategoryDao {

    int count();
}

```

```

        List<Category> findByCategoryname(String catname);

        List<Category> findAll(int page, int pagesize);

        List<Category> findAll();

        Category findById(int categoryId);

        void delete(int cateid) throws Exception;

        void update(Category category);

        void insert(Category category);
    }
    =====
    package vn.iotstar.Dao;

    import java.util.List;

    import javax.persistence.EntityManager;
    import javax.persistence.EntityTransaction;
    import javax.persistence.Query;
    import javax.persistence.TypedQuery;

    import vn.iotstar.JPAConfig.JPAConfig;
    import vn.iotstar.entity.Category;

    public class CategoryDaoImpl implements ICategoryDao {
        @Override
        public void insert(Category category) {
            EntityManager enma = JPAConfig.getEntityManager();
            EntityTransaction trans = enma.getTransaction();
            try {
                trans.begin();
                enma.persist(category);
                trans.commit();
            } catch (Exception e) {
                e.printStackTrace();
                trans.rollback();
                throw e;
            } finally {
                enma.close();
            }
        }
        @Override
        public void update(Category category) {
            EntityManager enma = JPAConfig.getEntityManager();
            EntityTransaction trans = enma.getTransaction();
            try {
                trans.begin();
                enma.merge(category);
                trans.commit();
            }

```

```

    } catch (Exception e) {
        e.printStackTrace();
        trans.rollback();
        throw e;
    }finally {
        enma.close();
    }
}
@Override
public void delete(int cateid) throws Exception {
    EntityManager enma = JPAConfig.getEntityManager();
    EntityTransaction trans = enma.getTransaction();
    try {
        trans.begin();
        Category category = enma.find(Category.class, cateid);
        if(category != null) {
            enma.remove(category);
        }else {
            throw new Exception("Không tìm thấy");
        }

        trans.commit();
    } catch (Exception e) {
        e.printStackTrace();
        trans.rollback();
        throw e;
    }finally {
        enma.close();
    }
}
@Override
public Category findById(int categoryId){
    EntityManager enma = JPAConfig.getEntityManager();
    Category category = enma.find(Category.class, categoryId);
    return category;
}
@Override
public List<Category> findAll(){
    EntityManager enma = JPAConfig.getEntityManager();
    TypedQuery<Category> query= enma.createNamedQuery("Category.findAll",
Category.class);
    return query.getResultList();
}
@Override
public List<Category> findAll(int page,int pagesize){
    EntityManager enma = JPAConfig.getEntityManager();
    TypedQuery<Category> query= enma.createNamedQuery("Category.findAll",
Category.class);
    query.setFirstResult(page*pagesize);
    query.setMaxResults(pagesize);
    return query.getResultList();
}
@Override
public List<Category> findByCategoryname(String catname){
    EntityManager enma = JPAConfig.getEntityManager();

```

```

        String jpql = "SELECT c FROM Category c WHERE c.catename like :catname";
        TypedQuery<Category> query= enma.createQuery(jpql, Category.class);
        query.setParameter("catename", "%" + catname + "%");

        return query.getResultList();
    }
    @Override
    public int count(){
        EntityManager enma = JPAConfig.getEntityManager();
        String jpql = "SELECT count(c) FROM Category c";
        Query query = enma.createQuery(jpql);
        return ((Long)query.getSingleResult()).intValue();
    }
}

```

Bước 4: xây dựng tầng service

```

package vn.iotstar.services;

import java.util.List;

import vn.iotstar.entity.Category;

public interface ICategoryService {
    void insert(Category category);
    void update(Category category);
    void delete(int cateid) throws Exception;
    Category findById(int cateid);
    List<Category> findAll();
    List<Category> findByCategoryname(String catname);
    List<Category> findAll(int page,int pagesize);
    int count();
}
=====
package vn.iotstar.services;

import java.util.List;

import vn.iotstar.Dao.CategoryDaoImpl;
import vn.iotstar.Dao.ICategoryDao;
import vn.iotstar.entity.Category;

public class CategoryServiceImpl implements ICategoryService {
    ICategoryDao categoryDao =new CategoryDaoImpl();

    @Override
    public void insert(Category category) {
        categoryDao.insert(category);
    }
    @Override
    public void update(Category category) {
        categoryDao.update(category);
    }
    @Override

```

```

    public void delete(int cateid) throws Exception {
        categoryDao.delete(cateid);
    }
    @Override
    public Category findById(int cateid) {
        return categoryDao.findById(cateid);
    }
    @Override
    public List<Category> findAll() {
        return categoryDao.findAll();
    }
    @Override
    public List<Category> findByCategoryname(String catname) {

        return categoryDao.findByCategoryname(catname);
    }
    @Override
    public List<Category> findAll(int page, int pagesize) {

        return categoryDao.findAll(page, pagesize);
    }
    @Override
    public int count() {

        return categoryDao.count();
    }
}

```

Bước 5: Viết hàm thư viện hỗ trợ chuyển đổi Object sang Json và ngược lại

```

package vn.iotstar.util;

import java.io.BufferedReader;
import java.io.IOException;

import com.fasterxml.jackson.databind.ObjectMapper;

public class HttpUtil {

    private String value;

    public HttpUtil (String value) {
        this.value = value;
    }

    public <T> T toModel(Class<T> tClass) {
        try {
            return new ObjectMapper().readValue(value, tClass);
        } catch (Exception e) {
            System.out.print(e.getMessage());
        }
        return null;
    }
}

```

```

        public static HttpUtil of (BufferedReader reader) {
            StringBuilder sb = new StringBuilder();
            try {
                String line;
                while ((line = reader.readLine()) != null) {
                    sb.append(line);
                }
            } catch (IOException e) {
                System.out.print(e.getMessage());
            }
            return new HttpUtil(sb.toString());
        }
    }
}

```

```

<!--
    https://mvnrepository.com/artifact/commons-beanutils/commons-beanutils -
->

    <dependency>
        <groupId>commons-beanutils</groupId>
        <artifactId>commons-beanutils</artifactId>
        <version>1.9.4</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/commons-io/commons-io -->
    <dependency>
        <groupId>commons-io</groupId>
        <artifactId>commons-io</artifactId>
        <version>2.11.0</version>
    </dependency>
=====
public class Constant {
    public static final String DIR = "E:\\uploads";
}

```

```

package vn.iotstar.util;

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.Part;

import org.apache.commons.io.FilenameUtils;

public class UploadUtils {
    public static String processUpload(String fieldName, HttpServletRequest req,
String storeFolder, String storeFilename) throws IOException, ServletException {
        Part filePart = req.getPart(fieldName);
        if(filePart == null || filePart.getSize()==0) {
            return "";
        }
    }
}

```



```

        if(storeFolder ==null) {
            storeFolder ="E:\\uploads";
        }
        if (storeFilename == null) {
            storeFilename =
Paths.get(filePart.getSubmittedFileName()).getFileName().toString();
        }else {
            storeFilename+="." +
FilenameUtils.getExtension(Paths.get(filePart.getSubmittedFileName()).getFileName().t
oString());
        }
        Path uploadPath = Paths.get(storeFolder);
        if(!Files.exists(uploadPath)) {
            Files.createDirectories(uploadPath);
        }

        filePart.write(Paths.get(uploadPath.toString(),storeFilename).toString());
        return storeFilename;
    }

    public static String processUploadFolderWeb(String fieldName,
HttpServletRequest req, String storeFolder, String storeFilename) throws IOException,
ServletException {
        Part filePart = req.getPart(fieldName);
        if(filePart == null || filePart.getSize()==0) {
            return "";
        }
        if(storeFolder ==null) {
            storeFolder ="/uploads";
        }
        if (storeFilename == null) {
            storeFilename =
Paths.get(filePart.getSubmittedFileName()).getFileName().toString();
        }else {
            storeFilename+="." +
FilenameUtils.getExtension(Paths.get(filePart.getSubmittedFileName()).getFileName().t
oString());
        }
        //lấy đường dẫn trong web
        String uploadFolder = req.getServletContext().getRealPath(storeFolder);

        Path uploadPath = Paths.get(uploadFolder);
        //hết lấy đường dẫn trong web
        if(!Files.exists(uploadPath)) {
            Files.createDirectories(uploadPath);
        }

        filePart.write(Paths.get(uploadPath.toString(),storeFilename).toString());
        return storeFilename;
    }
}

```

=====

Bước 6: Viết Controller API

```

package vn.iotstar.controllers.api;

import java.io.File;
import java.io.IOException;
import java.lang.reflect.InvocationTargetException;

import javax.inject.Inject;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.apache.commons.beanutils.BeanUtils;

import com.fasterxml.jackson.databind.ObjectMapper;

import vn.iotstar.entity.Category;

import vn.iotstar.services.ICategoryService;
import vn.iotstar.util.Constant;
import vn.iotstar.util.HttpUtil;
import vn.iotstar.util.UploadUtils;

@MultipartConfig
@WebServlet(urlPatterns = {"/api-admin-category"})
public class CategoryAPI extends HttpServlet {

    @Inject
    private ICategoryService categoryService;

    private static final long serialVersionUID = -915988021506484384L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        ObjectMapper mapper = new ObjectMapper();
        request.setCharacterEncoding("UTF-8");
        response.setContentType("application/json");
        mapper.writeValue(response.getOutputStream(),
categoryService.findAll());
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        ObjectMapper mapper = new ObjectMapper();
        request.setCharacterEncoding("UTF-8");
        response.setContentType("application/json");
        // khởi tạo đối tượng entity
        Category category = new Category();
        // sử dụng BeanUtils để tự lấy các name Field trên form
        // tên field phải trùng với entity
        try {
            BeanUtils.populate(category, request.getParameterMap());

```

```

    } catch (IllegalAccessException | InvocationTargetException e) {
        e.printStackTrace();
    }

    // xử lý hình ảnh
    String fileName = "" + System.currentTimeMillis();

    category.setIcon(UploadUtils.processUpload("icon", request,
Constant.DIR+"\\category\\", fileName));

    categoryService.insert(category);
    mapper.writeValue(response.getOutputStream(), category);
}

protected void doPut(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    ObjectMapper mapper = new ObjectMapper();
    request.setCharacterEncoding("UTF-8");
    response.setContentType("application/json");
    // lấy dữ liệu từ jsp bằng BeanUtils
    Category category = new Category();
    try {
        BeanUtils.populate(category, request.getParameterMap());
    } catch (IllegalAccessException | InvocationTargetException e) {
        e.printStackTrace();
    }
    // khởi tạo DAO
    Category oldcate = categoryService.findById(category.getCategoryID());
    // xử lý hình ảnh
    if (request.getPart("icon").getSize() == 0) {
        category.setIcon(oldcate.getIcon());
    } else {
        if (oldcate.getIcon() != null) {
            // XOA ANH CU DI
            String fileName = oldcate.getIcon();
            File file = new File(Constant.DIR+"\\category\\" + fileName);
            if (file.delete()) {
                System.out.println("Đã xóa thành công");
            } else {
                System.out.println(Constant.DIR+"\\category\\" + fileName);
            }
        }
        String fileName = "" + System.currentTimeMillis();

        category.setIcon(UploadUtils.processUpload("icon", request,
Constant.DIR+"\\category\\", fileName));

    }

    categoryService.update(category);
    mapper.writeValue(response.getOutputStream(), category);
}

```

```

protected void doDelete(HttpServletRequest request, HttpServletResponse
response)
    throws ServletException, IOException {
    ObjectMapper mapper = new ObjectMapper();
    request.setCharacterEncoding("UTF-8");
    response.setContentType("application/json");
    Category cateModel =
HttpUtil.of(request.getReader()).toModel(Category.class);
    try {
        categoryService.delete(cateModel.getCategoryID());
    } catch (Exception e) {
        e.printStackTrace();
    }
    mapper.writeValue(response.getOutputStream(), "{Đã xóa thành công}");
}
}

```

Bước 7: Kết quả GET

GET http://localhost:8080/Hello4/api-admin-category Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 1181 ms Size: 16.88 KB Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "icon": "1697424255323.png",
3   "products": [
4     {
5       "stoke": 20,
6       "imageLink": "https://canary.contesting.wish.com/api/webimage/60ee87ac5f321f54b6515aa5-large.jpg",
7       "description": "Chúng tôi luôn đưa đến cho khách hàng những sản phẩm nhập khẩu tốt nhất với quy trình sản xuất chuyên nghiệp:\r\n\r\n Hàng hóa nhập khẩu về VN được kiểm định chất
lượng.",
8       "productID": 2,
9       "sellerID": 1,
10      "amount": 90,
11      "productName": "Bộ quần áo cá sấu",
12      "price": 274990
13    },
14  ],

```

POST

POST http://localhost:8080/Hello4/api-admin-category Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> categoryName	Trung 1			
<input checked="" type="checkbox"/> icon	Ob2971bd-7098-4652-a677-c61aea36e032.webp			
Key	Value	Description		

Body Cookies Headers (5) Test Results Status: 200 OK Time: 5.42 s Size: 244 B Save as example

Pretty Raw Preview Visualize JSON

```

1 {
2   "icon": "1699926888732.webp",
3   "products": null,
4   "categoryID": 6041,
5   "categoryName": "Trung 1"
6 }

```

PUT

PUT ▼ http://localhost:8080/Hello4/api-admin-category Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> categoryName	Trung 2	
<input checked="" type="checkbox"/> icon	Select Files	
<input checked="" type="checkbox"/> categoryID	6041	
Key	Value	Description

Body Cookies Headers (5) Test Results Status: 200 OK Time: 938 ms Size: 244 B Save as example

Pretty Raw Preview Visualize JSON ▼ ⌵

```
1 {
2   "icon": "169992688732.webp",
3   "products": null,
4   "categoryID": 6041,
5   "categoryName": "Trung 2"
6 }
```

DELETE

DELETE ▼ http://localhost:8080/Hello4/api-admin-category Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON ▼ Beautify

```
1 {
2   \"categoryID\": \"6041\"
3 }
4 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 880 ms Size: 182 B Save as example

Pretty Raw Preview Visualize JSON ▼ ⌵

```
1 \"{Bỏ xóa thành công}\"
```

2. HƯỚNG DẪN CRUD CATEGORY TRÊN SERVLET VỚI THƯ VIỆN GSON

Bước 1: Thêm Thư viện

```
<!-- Gson -->
<dependency>
  <groupId>com.google.code.gson</groupId>
  <artifactId>gson</artifactId>
  <version>2.8.6</version>
</dependency>
<!-- End Gson -->
```

Bước 2: Xây dựng Entity CATEGORY, PRODUCT xem lại bước 2 của mục 1

=====

Bước 3: xây dựng tầng DAO xem lại bước 3 của mục 1

Bước 4: xây dựng tầng service xem lại bước 4 của mục 1

=====

Bước 6: Viết Controller API

```
package vn.iotstar.controllers.api;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
```

```

import java.lang.reflect.InvocationTargetException;
import java.util.List;

import javax.inject.Inject;
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.beanutils.BeanUtils;
import com.google.gson.Gson;
import vn.iotstar.entity.Category;
import vn.iotstar.services.ICategoryService;
import vn.iotstar.util.Constant;

import vn.iotstar.util.UploadUtils;

@MultipartConfig
@WebServlet("/api-admin-cate")
public class CategoryAPIController extends HttpServlet {

    @Inject
    ICategoryService categoryService;

    private static final long serialVersionUID = 1L;

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        resp.setContentType("application/json");
        resp.setCharacterEncoding("UTF-8");
        // String json =
        //
        '{"name':'Trung','icon':'category/687f3967b7c2fe6a134a2c11894eea4b_tn.png'}";

        // khởi tạo đối tượng entity
        Category category = new Category();
        // sử dụng BeanUtils để tự lấy các name Field trên form
        // tên field phải trùng với entity

        try {
            BeanUtils.populate(category, req.getParameterMap());
        } catch (IllegalAccessException | InvocationTargetException e) {

            e.printStackTrace();
        }

        // xử lý hình ảnh
        String fileName = "" + System.currentTimeMillis();
        category.setIcon(UploadUtils.processUpload("icon", req,
Constant.DIR+"\\category\\", fileName));

```

```

Gson gson = new Gson(); // Khởi tạo đối tượng Gson
// gọi hàm fromJson để chuyển từ String sang Object
String cateJson = gson.toJson(category);
// gọi hàm insert trong Service để thêm dữ liệu đối tượng vào database

Category cate = gson.fromJson(cateJson, Category.class);

categoryService.insert(cate);
// Viết thông báo kết quả
PrintWriter out = resp.getWriter();
out.println("Đã thêm thành công");
out.println(cateJson);
out.close();
}

@Override
protected void doPut(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
    resp.setContentType("application/json");
    resp.setCharacterEncoding("UTF-8");
    // lấy dữ liệu từ jsp bằng BeanUtils
    Category category = new Category();
    try {
        BeanUtils.populate(category, req.getParameterMap());
    } catch (IllegalAccessException | InvocationTargetException e) {
        e.printStackTrace();
    }
    // khởi tạo DAO
    Category oldcate = categoryService.findById(category.getCategoryID());
    // xử lý hình ảnh
    if (req.getPart("icon").getSize() == 0) {
        category.setIcon(oldcate.getIcon());
    } else {
        if (oldcate.getIcon() != null) {
            // XOA ANH CU DI
            String fileName = oldcate.getIcon();
            File file = new File(Constant.DIR+"\\category\\" +
fileName);

            if (file.delete()) {
                System.out.println("Đã xóa thành công");
            } else {
                System.out.println(Constant.DIR+"\\category\\" +
fileName);
            }
        }
        String fileName = "" + System.currentTimeMillis();
        category.setIcon(UploadUtils.processUpload("icon", req,
Constant.DIR+"\\category\\", fileName));
    }
    Gson gson = new Gson(); // Khởi tạo đối tượng Gson
    // gọi hàm fromJson để chuyển từ String sang Object
    String cateJson = gson.toJson(category);
    // gọi hàm insert trong Service để thêm dữ liệu đối tượng vào database

```

```

        Category cate = gson.fromJson(cateJson, Category.class);

        // gọi hàm edit trong Service để cập nhật dữ liệu đối tượng vào database
        categoryService.update(cate);
        // Viết thông báo kết quả
        PrintWriter out = resp.getWriter();
        out.println("Đã sửa thành công");
        out.println(cateJson);
        out.close();
    }

    @Override
    protected void doDelete(HttpServletRequest req, HttpServletResponse resp)
    throws ServletException, IOException {
        resp.setContentType("application/json");
        resp.setCharacterEncoding("UTF-8");
        try {
            // String json =
            //
            "{ 'name': 'Trung', 'icon': 'category/687f3967b7c2fe6a134a2c11894eea4b_tn.png' }";
            Gson gson = new Gson(); // Khởi tạo đối tượng Gson
            // gọi hàm fromJson để chuyển từ String sang Object
            Category cateJson = gson.fromJson(req.getReader(),
            Category.class);

            // gọi hàm insert trong Service để thêm dữ liệu đối tượng
            vào database

            categoryService.delete(cateJson.getCategoryID());
            // Viết thông báo kết quả
            PrintWriter out = resp.getWriter();
            out.println("Đã xóa thành công");
            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

Bước 7: Kết quả

POST

POST http://localhost:8080/Hello4/api-admin-cate

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description
<input checked="" type="checkbox"/> categoryName	Trung 8	
<input checked="" type="checkbox"/> icon	2.PNG	
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 270 ms Size: 265 B Save as example

Pretty Raw Preview Visualize JSON

```

1  Đã thêm thành công
2  {
3      "CategoryID": 0,
4      "CategoryName": "Trung 8",
5      "Icon": "1699930856595.PNG"
6  }

```


PUT

PUT

http://localhost:8080/Hello4/api-admin-cate

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Key	Value	Description	
<input checked="" type="checkbox"/> categoryName	Trung 7		
<input checked="" type="checkbox"/> icon	2.PNG		
<input checked="" type="checkbox"/> categoryID	6051		
Key	Value	Description	

Body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

Đã sửa thành công

2

{

3

"CategoryID": 6051,

4

"CategoryName": "Trung 7",

5

"icon": "1699930780159.PNG"

6

}

Status: 200 OK

Time: 807 ms

Size: 268 B

Save as example

DELETE

DELETE

http://localhost:8080/Hello4/api-admin-cate

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"CategoryID": "6051"

3

}

body

Cookies

Headers (5)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

Đã xóa thành công

2

Status: 200 OK

Time: 4.44 s

Size: 194 B

Save as example

3. HƯỚNG DẪN CRUD CATEGORY TRÊN SPRING BOOT