|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例名称 | | 上传商品 | | | |
| 测试类型 | | 单元测试 | | | |
| 测试目的 | | 在客户端上传商品至数据库，保证数据的完整性 | | | |
| 测试方法 | | 在本地打开客户端上传商品 | | | |
| 测试用例设计原则 | | 条件组合覆盖 | | | |
| 正常测试情况 | | | | | |
| 编号 | 测试数据 | 数据描述 | 预期结果 | 实际测试结果 | |
| 1 | name:空 | 商品名称为空 | 错误提示：商品名不能为空 | 与预期结果相同 | |
| amount:”red,s,100,200” | 字段不符合输入规定 |
| raw\_price:"a3" | 字段不符合输入规定 |
| 2 | name:空 | 商品名称为空 | 错误提示：商品名不能为空 | 与预期结果相同 | |
| amount:”red,s,100,200” | 字段不符合输入规定 |
| raw\_price:"500" | 字段符合要求 |
| 3 | name:空 | 商品名称为空 | 错误提示：商品名不能为空 | 与预期结果相同 | |
| amount:”red,s,100” | 字段符合要求 |
| raw\_price:"a3" | 字段不符合输入规定 |
| 4 | name:空 | 商品名称为空 | 错误提示：商品名不能为空 | 与预期结果相同 | |
| amount:”red,s,100” | 字段符合要求 |
| raw\_price:"500" | 字段符合要求 |
| 5 | name:"测试商品1" | 字段符合要求 | 错误提示：请输入合法数据 | 与预期结果相同 | |
| amount:”red,s,100,200” | 字段不符合输入规定 |
| raw\_price:"a3" | 字段不符合输入规定 |
| 6 | name:"测试商品1" | 字段符合要求 | 错误提示：请输入合法数据 | 与预期结果相同 | |
| amount:”red,s,100,200” | 字段不符合输入规定 |
| raw\_price:"500" | 字段符合要求 |
| 7 | name:"测试商品1" | 字段符合要求 | 错误提示：请输入合法数据 | 与预期结果相同 | |
| amount:”red,s,100” | 字段符合要求 |
| raw\_price:"a3" | 字段不符合输入规定 |
| 8 | name:"测试商品1" | 字段符合要求 | 添加成功 | 与预期结果相同 | |
| amount:”red,s,100” | 字段符合要求 |
| raw\_price:"500" | 字段符合要求 |
| 测试结论 | |  |  |  |  |
| 状态 | | 正常 |  |  |  |
| 错误说明 | | 无 | | | |
| 错误重现说明 | | 无 | | | |
| 错误重现 | | 无 | | | |
| 测试人员 | | 陈董锴 | | 测试时间 | 2017/12/31 |

**被测试方法的代码：**

String name = this.edtName.getText();

if(name.equals("")||name==null) {

JOptionPane.showMessageDialog(null, "字段不能为空", "错误", JOptionPane.ERROR\_MESSAGE);

return;

}

String Desc = this.edtDesc.getText();

String Class = this.edtClass.getText();

double RPrice = 0, SPrice = 0, APrice = 0;

try {

RPrice = Double.parseDouble(this.edtRPrice.getText());

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "请输入合法数据", "错误", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

SPrice = Double.parseDouble(this.edtSPrice.getText());

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "请输入合法数据", "错误", JOptionPane.ERROR\_MESSAGE);

return;

}

try {

APrice = Double.parseDouble(this.edtAPrice.getText());

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "请输入合法数据", "错误", JOptionPane.ERROR\_MESSAGE);

return;

}

String TimeEnd = this.edtTimeEnd.getText();

String s = this.edtAmount.getText();

String[] str=s.split("@|,| ");

//System.out.println(str.length);

ArrayList<Integer> x= new ArrayList<Integer>();//amount

ArrayList<String> y=new ArrayList<String>();//color

ArrayList<String> z=new ArrayList<String>();//size

try {

for(int i=0; i<str.length; i+=3){

y.add(str[i]);

z.add(str[i+1]);

x.add(Integer.parseInt(str[i+2]));

System.out.println(str[i]+" "+str[i+1]+" "+str[i+2]);

}

int amount[]=new int[x.size()];

String[] Color =new String[y.size()];

String[] Size =new String[z.size()];

for(int i = 0; i < x.size();i++){

Color[i]=y.get(i);

Size[i]=z.get(i);

amount[i]=x.get(i);

}

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "请输入合法数据", "错误", JOptionPane.ERROR\_MESSAGE);

return;

}

String temp = edtSrc.getText();

String[] ans=temp.split("\\\\");

String src = "product\\"+ans[ans.length-1];

String id = Integer.toHexString((int)System.currentTimeMillis());

Products products = new Products();

products.setId(id);

products.setName(name);

products.setDescription(Desc);

products.setClass\_belong(Class);

products.setSize(z);

products.setColor(y);

products.setRaw\_price(RPrice);

products.setSale\_price(SPrice);

products.setAgent\_price(APrice);

products.setTime\_import(new Date());

products.setTime\_end\_sale(TimeEnd);

products.setAmount(x);

products.setSrc(src);

try{

Upload.upload(temp,src);

}

catch (Exception ex) {

JOptionPane.showMessageDialog(null, "图片上传失败", "错误", JOptionPane.ERROR\_MESSAGE);

return;

}

(new ProductsManager()).addProducts(products);

**用例选择原因：**

从代码中可以看出，有判断的字段有name,amount,Raw\_price,Sale\_price, Agent\_price;

但是Raw\_price,Sale\_price, Agent\_price的判断是等价的，所以我们只对name,amount,Raw\_price这三个字段设计测试用例，一共是8组数据。

**测试工具：**Juint

**测试代码：**

public void testAddProducts() throws Exception {

MongoClient mongoClient = new MongoClient( "localhost" , 27017 );

MongoDatabase mongoDatabase = mongoClient.getDatabase("SE2017");

MongoCollection<Document> collection = mongoDatabase.getCollection("products");

String name = "测试商品"+Integer.toHexString((int)System.currentTimeMillis());

String Desc = "xxx";

String Class = "xxx";

double RPrice = 500, SPrice = 1000, APrice = 700;

String TimeEnd = "2018/3/1";

String s ="red,s,100,red,m,200,red,l,150,blue,s,200,blue,m,200";

String[] str=s.split("@|,| ");

System.out.println(str.length);

ArrayList<Integer> x= new ArrayList<Integer>();//amount

ArrayList<String> y=new ArrayList<String>();//color

ArrayList<String> z=new ArrayList<String>();//size

for(int i=0; i<str.length; i+=3){

y.add(str[i]);

z.add(str[i+1]);

x.add(Integer.parseInt(str[i+2]));

}

int amount[]=new int[x.size()];

String[] Color =new String[y.size()];

String[] Size =new String[z.size()];

for(int i = 0; i < x.size();i++){

Color[i]=y.get(i);

Size[i]=z.get(i);

amount[i]=x.get(i);

}

String temp = "100";

String[] ans=temp.split("\\\\");

String src = "product\\"+ans[ans.length-1];

String id = Integer.toHexString((int)System.currentTimeMillis());

Document document = new Document("\_id", id)

.append("name", name)

.append("description",Desc)

.append("class\_belong", Class )

.append("size", z)

.append("color", y)

.append("raw\_price", RPrice )

.append("sale\_price", SPrice)

.append("agent\_price", APrice)

.append("time\_import", new Date() )

.append("time\_end\_sale", TimeEnd)

.append("amount", x)

.append("src",src);

collection.insertOne(document);

System.out.println("文档插入成功");

}