

Log in / create account

Openbravo.comIssuesForgeUniversity

View source | Discuss this page | Page history | Printable version

POS/2.30/HowTos/How To Add a New Field in Product Panel

Languages:English | Translate this article...

Contents

1 How to add a new field in Product panel

1.1 Objective

1.2 Process

1.2.1 Step 1

1.2.2 Step 2

1.2.3 Step 3

1.2.4 Step 4

1.2.5 Step 5

1.3 Result

How to add a new field in Product panel

Objective

This how to describes how to add a new information for products and allows to edit it through the products panel of Openbravo POS.

Process

Step 1

• Create a new column in PRODUCTS table

ALTER TABLE PRODUCTS ADD COLUMN FLAG BOOLEAN DEFAULT '0';

Step 2

Edit ProductEditor class (com.openbravo.pos.inventory)

• Create a JCheckBox (jcbFlag)

• In the constructor add the listener

jcbFlag.addChangeListener(e){try{

• Add this line to writeValueEOPF method

jcbFlag.setSelected(false);

• Add this line to writeValueInsert method

jcbFlag.setSelected(false);

• Add this line to writeValueDelete method

jcbFlag.setSelected(!Boolean.valueOf(l7)).booleanValue();

jcbFlag.setSelected(false);

• Add this line to writeValueEdit method

jcbFlag.setSelected(!Boolean.valueOf(l7)).booleanValue();

• Add this line to createValue method

myprod17 = Boolean.valueOf(jcbFlag.isSelected());

Step 3

Edit DataLogicSales class (com.openbravo.pos.forms)

• Add the new column in the constructor

new Field("FLAG", Datos.BOOLEAN, Formats.BOOLEAN)

• Change this methods

public final ProductInfoExt getProductInfo(String id) throws BasicException {
return (ProductInfoExt) new PreparedSentence(
"SELECT ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, TAXCAT, CATEGORY, ATTRIBUTESET_ID, IMAGE, ATTRIBUTES, FLAG " +
"FROM PRODUCTS WHERE ID = ?"
).SerializerWriteString.INSTANCE
ProductInfoExt.getSerializerRead()).find(id);
}
public final ProductInfoExt getProductInfoByCode(String sCode) throws BasicException {
return (ProductInfoExt) new PreparedSentence(
"SELECT ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, TAXCAT, CATEGORY, ATTRIBUTESET_ID, IMAGE, ATTRIBUTES, FLAG " +
"FROM PRODUCTS WHERE CODE = ?"
).SerializerWriteString.INSTANCE
ProductInfoExt.getSerializerRead()).find(sCode);
}
public final ProductInfoExt getProductInfoByReference(String sReference) throws BasicException {
return (ProductInfoExt) new PreparedSentence(
"SELECT ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, TAXCAT, CATEGORY, ATTRIBUTESET_ID, IMAGE, ATTRIBUTES, FLAG " +
"FROM PRODUCTS WHERE REFERENCE = ?"
).SerializerWriteString.INSTANCE
ProductInfoExt.getSerializerRead()).find(sReference);
}
public List<ProductInfoExt> getProductCatalog(String category) throws BasicException {
return new PreparedSentence(
"SELECT P.ID, P.REFERENCE, P.CODE, P.NAME, P.ISCOM, P.ISSCALE, P.PRICEBUY, P.PRICESELL, P.TAXCAT, P.CATEGORY, P.ATTRIBUTESET_ID, P.IMAGE, P.ATTRIBUTES, P.FLAG " +
"FROM PRODUCTS P, PRODUCTS_CAT O WHERE P.ID = O.PRODUCT AND P.CATEGORY = ? "
"ORDER BY O.CATORDER, P.NAME"
).SerializerWriteString.INSTANCE
ProductInfoExt.getSerializerRead()).list(category);
}
public List<ProductInfoExt> getProductComments(String id) throws BasicException {
return new PreparedSentence(
"SELECT P.ID, P.REFERENCE, P.CODE, P.NAME, P.ISCOM, P.ISSCALE, P.PRICEBUY, P.PRICESELL, P.TAXCAT, P.CATEGORY, P.ATTRIBUTESET_ID, P.IMAGE, P.ATTRIBUTES, P.FLAG " +
"FROM PRODUCTS P, PRODUCTS_CAT O, PRODUCTS_COM M WHERE P.ID = O.PRODUCT AND P.ID = M.PRODUCT2 AND M.PRODUCT = ? "
"AND P.ISCOM = " + s.DB.TRUE() + " "
"ORDER BY O.CATORDER, P.NAME"
).SerializerWriteString.INSTANCE
ProductInfoExt.getSerializerRead()).list(id);
}
public final SentenceList getProductList() {
return new StaticSentence(
new QBBuilder(
"SELECT ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, TAXCAT, CATEGORY, ATTRIBUTESET_ID, IMAGE, ATTRIBUTES, FLAG " +
"FROM PRODUCTS WHERE ?(QB_FILTER) ORDER BY REFERENCE", new String[] {"NAME", "PRICEBUY", "PRICESELL", "CATEGORY", "CODE"})
).SerializerWriteBasic(new Datos[] {Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.STRING})
ProductInfoExt.getSerializerRead());
}
// Products list
public SentenceList getProductListNormal() {
return new StaticSentence(
new QBBuilder(
"SELECT ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, TAXCAT, CATEGORY, ATTRIBUTESET_ID, IMAGE, ATTRIBUTES, FLAG " +
"FROM PRODUCTS WHERE ISCOM = " + s.DB.FALSE() + " AND ?(QB_FILTER) ORDER BY REFERENCE", new String[] {"NAME", "PRICEBUY", "PRICESELL", "CATEGORY", "CODE"})
).SerializerWriteBasic(new Datos[] {Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.DOUBLE, Datos.OBJECT, Datos.DOUBLE, Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.STRING})
ProductInfoExt.getSerializerRead());
}
//Auxiliar list for a filter
public SentenceList getProductListAuxiliar() {
return new StaticSentence(
new QBBuilder(
"SELECT ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, TAXCAT, CATEGORY, ATTRIBUTESET_ID, IMAGE, ATTRIBUTES, FLAG " +
"FROM PRODUCTS WHERE ISCOM = " + s.DB.TRUE() + " AND ?(QB_FILTER) ORDER BY REFERENCE", new String[] {"NAME", "PRICEBUY", "PRICESELL", "CATEGORY", "CODE"})
).SerializerWriteBasic(new Datos[] {Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.DOUBLE, Datos.OBJECT, Datos.DOUBLE, Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.STRING})
ProductInfoExt.getSerializerRead());
}
public final SentenceList getProductCatQBF() {
return new StaticSentence(
new QBBuilder(
"SELECT P.ID, P.REFERENCE, P.CODE, P.NAME, P.ISCOM, P.ISSCALE, P.PRICEBUY, P.PRICESELL, P.CATEGORY, P.TAXCAT, P.ATTRIBUTESET_ID, P.IMAGE, P.STOCKCOST, P.STOCKVOLUME, CASE WHEN C.PRODUCT IS NULL THEN " + s.DB.FALSE() + " ELSE " + s.DB.TRUE() + " END, C.CATORDER, P.ATTRIBUTES, P.FLAG " +
"FROM PRODUCTS P LEFT OUTER JOIN PRODUCTS_CAT C ON P.ID = C.PRODUCT " +
"WHERE ?(QB_FILTER) " +
"ORDER BY P.REFERENCE", new String[] {"P.NAME", "P.PRICEBUY", "P.PRICESELL", "P.CATEGORY", "P.CODE"})
).SerializerWriteBasic(new Datos[] {Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.DOUBLE, Datos.OBJECT, Datos.DOUBLE, Datos.OBJECT, Datos.STRING, Datos.OBJECT, Datos.STRING})
productsShow.getSerializerRead());
}
public final SentenceExec getProductCatInsert() {
return new SentenceExec(Transactions() {
public List execInTransaction(Object params) throws BasicException {
Object[] values = (Object[]) params;
int i = new PreparedSentence(
"INSERT INTO PRODUCTS (ID, REFERENCE, CODE, NAME, ISCOM, ISSCALE, PRICEBUY, PRICESELL, CATEGORY, TAXCAT, ATTRIBUTESET_ID, IMAGE, STOCKCOST, STOCKVOLUME, ATTRIBUTES,FLAG) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
).SerializerWriteBasicExt(productsRow.getDatas()), new int[] {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16,17})).exec(params);
if (i > 0 && (Boolean.valueOf(i)>14).booleanValue()) {
return new PreparedSentence(
"INSERT INTO PRODUCTS_CAT (PRODUCT, CATORDER) VALUES (?, ?)"
).SerializerWriteBasicExt(productsRow.getDatas()), new int[] {0, 15})).exec(params);
} else {

1 of 2

3/21/20, 7:49 PM

Category: Code Snippets POS