

How to create products

by the [OSOE Project](#).

```
<span metal:use-macro="container/Zuite_viewTestMacroLibrary/macros/init_test_environment">init</span>
```

```
openAndWait                                ${base_url}/Zuite_setUpProductTest
```

```
verifyTextPresent                          Init Ok
```

```
<span metal:use-macro="container/Zuite_viewTestMacroLibrary/macros/wait_for_activities"> Wait for activities</span>
```

```
<span metal:use-macro="container/Zuite_viewTestMacroLibrary/macros/login_as_functional_test_user">Login as Functional Test User</span>
```

This guide will teach you:

- How to create products/services
- How to validate a product/service
- How to create a complex product

In this tutorial, you will learn how to create and manage products and services in ERP5. This process has a similar structure as previous three tutorials for persons and organisations, which means you can try doing it by yourself or follow us along. This is the same three steps procedure :

Create new document.

Fill the different tabs with appropriate characteristics.

Validate your data.

Scenario



As electronics specialist, the core of your business is to sell high quality components, like this LED for example, which is your star product

We created an example to help you understand the different steps of product creation. Your company is an electronic component supplier/specialist, selling LED lights, which are little amps for electronic equipment.

We gathered some hypothetical characteristics, which will help you fill in the blanks, but please notice that those characteristics are not all in logical order, and you will have to sort them.

Characteristics

Title : Red Led

Reference : RL01

Electronic component

Taxable

Purchase price : 0,5€

Sales price: 1€

Validity : 01/01/2010

Supplier: EC China

Go to the « new » tab



You are now familiar with creating new objects in ERP5 as it was covered in the first several tutorials. Product creation works the same way :

First reach the “new” tab of the home menu which helps you create objects in ERP5. To reach the home menu, click on “ERP5” on the bread crumb.

As you can see on the screenshot, the only difference is that we are now using a new module called PDM : Product Data Management. With this module, you will be able to manage your commercial offers based on products and services. You will find further details in this tutorial and by browsing the application.

Click on product



To create a product, simply **click on “product”** .

This will create a new product, save it, and open the related product sheet, from which you'll be able to manage the details of the product, as described below.

```
selectAndWai //select[@name="select_modul Products
t           e"]

selectAndWai //select[@name="select_action" Add Product
t           ]

verifyText   //div[@id="transition_message" Object created.
]

verifyText   //li[@class="selected"]/a/span View
```

Fill the “View” tab



First, please fill the “view” tab, the same way we did for persons and organisations.

Look on the picture or in the first pages of this tutorial to find the details of the “Red Led”, and try to fill out the “view” tab, which is first step in the detail edition, because not all characteristics belong to this tab.

Note that since this product doesn't have any variation of size or price, **in the field "Individual Variation Base Category List", we can not choose "Variation". To do so, please click on the option "Variation", then hold "Ctrl" and click on "Variation" again** . When you are done, save your data, then you will see the result. Some fields will remain blank, this is not a problem. If a field is compulsory, ERP5 won't accept to save your data until you entered it.

This step will create a new product, save it, and open the related product sheet, from which you'll be able to manage the other details of the product, as described below.

Note: If the product has various purchase/sale prices or various sizes, e.g, a book with two different formats (big and small format), the price will be different, then you will have to choose "Variation" in the "Individual Variation Base Category List ", so you can set different prices in "Purchase" and "Sale" tabs later. We will show you examples in the next part "Complex products-Variations" in this tutorial.

```
type          //input[@name="field_my_title"]          ZUITE-TEST-PRODUCT-PRODU
001

type          //input[@name="field_my_reference"]      RL 01

select       //select[@name="field_my_base_contribution_list"]  Taxable

select       //select[@name="subfield_field_my_quantity_unit_list_def
ault:list"]  Unit/Piece

clickAndWait //button[@name="Base_edit:method"]
```

Fill the "Purchase" information



With the characteristics given in the example, we are able to fill out 5 fields of the “Purchase” tab. The reason for this is simple, ERP5 divides the data until it has reached its canonical form, which means that “Purchase price : 0.5€/piece” carries 3 values rather than 2 as it may seem: Each piece (1) is sold 0.5 (2) Euro (3).

We recommend you to always type commentaries about your products, and other data stored in ERP5.

```

verifyValue //input[@name="field_my_purchase_supply_line_source_title"] ZUITE-TEST-PRODUCT-ORGANISATION 001
verifyValue //input[@name="subfield_field_my_purchase_supply_line_start_date_range_max_year"] 2014
verifyValue //input[@name="subfield_field_my_purchase_supply_line_start_date_range_max_month"] 10
verifyValue //input[@name="subfield_field_my_purchase_supply_line_start_date_range_max_day"] 01
verifyValue //input[@name="field_my_purchase_supply_line_base_price"] 0.50
verifyValue //input[@name="field_my_purchase_supply_line_priced_quantity"] 1.0

```

Fill the "Sale" information



Now that we have filled correctly the purchase information, it is time to give details about your selling price, by **filling the "Sale" tab**.

Please, click at **"Sale" tab**, and try to fill the form with the following details :

Example : You sell your LED at a price of 1€ / piece. The sales price shall remain stable until 1st of January 2014. Many companies are used to purchasing this product from you.

There are four other fields to fill about the sales price, as shown on the screenshot : "Sales price", "currency", "Priced quantity" and "Price validity".

Note that the "Recipient" field has been left blank. This field is only used if the product is sold to one customer exclusively. As the Red LED is our star product and is sold to many different customers, this field remains blank.

```

clickAndWait link=Sale
type //input[@name="field_my_sale_supply_line_base_price"] 1
type //input[@name="field_my_sale_supply_line_priced_quantity"] 1.0
select //select[@name="field_my_sale_supply_line_price_currency"] EUR
clickAndWait //button[@name="Base_edit:method"]

```

About the “Stock” tab



Managing stock levels is another crucial business activity, and ERP5 allows you to track it smoothly thanks to the “inventory” function. As we will demonstrate in another tutorial, you can make an inventory to establish your stock levels for each product. If you do it correctly, your stock will be kept up to date by ERP5. You can find **this very information in the “Stock” tab of your product** . Each time you purchase or sell this product, ERP5 automatically updated its stock level.

Validate



Now that you have defined your product, it is time to validate it.

To do so, please follow the same steps as to validate a person or an organisation. The goal of this validation is to inform your co workers that the information you provide can be taken for granted. If you are not sure about the data you entered, do not validate, but rather leave it as draft to avoid confusion!

selectAndWait	//select[@name="select_action"]	Validate
clickAndWait	//button[@id="dialog_submit_button"]	
verifyText	//li[@class="selected"]/a/span	Sale
verifyText	//div[@id="transition_message"]	Status changed.
clickAndWait	link=View	
verifyText	//div[@title="The current state of the document."]/div[1]	Validated
storeLocation	product	

 Wait for activities

Complex products-Variations (1): Individual Variation Base Category List



In ERP5, you can create products that are much more complex than standard products we have just created. For example, you might need to enter an unique reference, to store different product variations. We will see here the example of a book, that is available in different formats, a big and a small format. Of

course, the prices are different, be it the purchase or sale price. But we would like to have only one reference for this product, displaying both the big and small format with their respective information such as prices and stock level.

When you create the new product, in the "View" tab, **choose "Variation" in the "Individual Variation Base Category List"** . When this is done, you can save and jump to next page.

```
openAndWait  ${product}
t
```

Complex products-Variations (2): Add Product Individual Variation



In this second step, we will add Product individual variations. In order to do this, **open the Action item** list and **choose "Add Product Individual Variation"** . A new document will be created for this variation and you will be redirected to it in order to edit it.

```
selectAndWait //select[@name="select_action Add Product Individual
t           "]          Variation
```

Complex products-Variations (3): Name your variation(s)



This is your product individual variation document that has been created. You can **set a name** to it and **save** when you are done.

After you are finished, **go back to the product document** by clicking on your product name in the Bread Crumb. Repeat these two steps to create another product individual variation document "Small Format".

```
type           //input[@name="field_my_title"]           ZUITE-PRODUCT-VARIATION-001
```

```
clickAndWait //button[@name="Base_edit:method"]
```

```
clickAndWait link=ZUITE-TEST-PRODUCT-PRODUCT-
001
```

```
selectAndWait //select[@name="select_action"]           Add Product Individual Variation
t
```

```
type           //input[@name='field_my_title']           ZUITE-PRODUCT-VARIATION-002
```

```
clickAndWait //button[@name="Base_edit:method"]
```

Go back to your product



Here we are back on the Product document. As you can see on the bottom of the document, a **list of Product Individual Variations is displayed** . You can click on any of them if you want to edit their name. But if you want to set their purchase and sales prices, please jump to next page of this tutorial.

```
clickAndWait link=ZUITE-TEST-PRODUCT-PRODUCT-001
```

```
verifyValue //input[@name='field_my_title'] ZUITE-TEST-PRODUCT-PRODUCT-001
```

```
verifyText //span[@class="listbox-current-page-total-number"] 2 records
```

Set the prices (1)



We will now see how to set the Purchase price of these product individual variations.

As you can see on the screenshot, the variations you created in the previous steps are not displayed here. Indeed, there is a step to take in order for you to be able to set the prices of those variations. Jump to next page.

```
clickAndWait link=Purchase
```

Set the prices (2)



In the blue area, **click on variation** and **click the Save button** . When you are done, jump to next page.

```
select //select[@name='field_my_purchase_supply_line_p_variation_base_category_variation_list'] Variation
```

```
clickAndWait //button[@name="Base_edit:method"]
```

Set the prices (3)



You can see in the red area that the two variations we have created in this tutorial are now displayed on the bottom of the page. You can now set the **purchase price** of each variation in the appropriate field, and **click the Save button** when you are done.

```

type          //input[@name="field_matrixbox_base_price_cell_0_0_0"] 10

type          //input[@name="field_matrixbox_base_price_cell_1_0_0"] 20

clickAndWait  //button[@name="Base_edit:method"]

verifyTextPresent Variation/ZUITE-PRODUCT-VARIATION-001

verifyTextPresent Variation/ZUITE-PRODUCT-VARIATION-002

```

Set the prices (4)



In order to define the Sale Price of your variations, **click on the Sale tab and take the same steps as for the Purchase price** .Do not forget to save your work when a document is modified.

```

clickAndWait link=Sale

select        //select[@name='field_my_sale_supply_line_p_variation_base_category_list'] Variation

clickAndWait //button[@name="Base_edit:method"]

type          //input[@name="field_matrixbox_base_price_cell_0_0_0"] 15

type          //input[@name="field_matrixbox_base_price_cell_1_0_0"] 25

clickAndWait //button[@name="Base_edit:method"]

```

Complex prices–Quantity steps (1)



ERP5 also manages Price Quantity Steps. This feature is useful if the price of a product varies with the ordered or sold quantity. In this example, we will set 3 different quantity steps for the purchase and the sale price, whether the quantity is less than 50 units, between 50 and 100 units and more than 100 units.

The Quantity step field will help you do this operation. Enter the different quantity steps you need in the Quantity step field, separating each of them by the Return Key of your keyboard. When you are done, click the Save button and jump to next page.

```
type //textarea[@name="field_my_sale_supply_line_quantity_step 50
_list"]

clickAndWait //button[@name="Base_edit:method"]
t
```

Complex prices–Quantity steps (2)



As we are using here the product for which we created individual variations, the Quantity steps will be applied to those different variations. As you can see on the picture, for each variation, three price fields are now available. The first column is the unit price for an order of less than 50 pieces, the second column is the unit price for an order between 50 and 100 units, and the column on the right is the unit price for an order of more than 100 pieces. Each line stands for a product individual variation, in our example, big format and small format. Jump to next page.

```
verifyTextPresent Quantity < 50.0

verifyTextPresent 50.0 <=
nt Quantity
```

Complex prices–Quantity steps (3)



Enter all the unit prices for your variations and quantity steps , and when you are done, **click the Save button** .

This was the very last step of this tutorial, you should now be able to create and manage products, including product individual variations and quantity steps. We would recommend to be very careful when editing a product, for **the information borne by the product documents will be automatically imported to the orders related to these products** , which will help you gain much time!

```
type //input[@name="field_matrixbox_base_price_cell_0_0 15
_0"]

type //input[@name="field_matrixbox_base_price_cell_0_1 14
_0"]

type //input[@name="field_matrixbox_base_price_cell_1_0 25
_0"]

type //input[@name="field_matrixbox_base_price_cell_1_1 24
```

```

        _0"]

clickAndWait //button[@name="Base_edit:method"]
t

verifyValue //input[@name="field_matrixbox_base_price_cell_0_0 15.00
_0"]

verifyValue //input[@name="field_matrixbox_base_price_cell_0_1 14.00
_0"]

verifyValue //input[@name="field_matrixbox_base_price_cell_1_0 25.00
_0"]

verifyValue //input[@name="field_matrixbox_base_price_cell_1_1 24.00
_0"]

```

Check you the Products creation

Now it is time to check the products on Product Module.

```

open          ${base_url}/view

selectAndWait //select[@name="select_module"] Products

verifyText    //div[@id="breadcrumb"]/a[1]    ERP5

verifyText    //div[@id="breadcrumb"]/a[2]    Products

<tal:block tal:content="structure python: here.Zuite_generateListboxBrowsing(listbox_selection=(['title',
'ZUITE-TEST-PRODUCT-PRODUCT-001']), enter_object=True)"/>

verifyText    //div[@id="breadcrumb"]/a[3]    ZUITE-TEST-PRODUCT-PRODUCT-001

<span metal:use-macro="container/Zuite_viewTestMacroLibrary/macros/login_as_manager">Login As
Manager</span>

openAndWait    $
                {base_url}/Zuite_tearDownProdu
                ctTest

verifyTextPresent    Clean Ok

<span metal:use-macro="container/Zuite_viewTestMacroLibrary/macros/wait_for_activities"> Wait for
activities </span>

```