Odoo module's documentation

Table des Matières

Table des matières

Documentation

1 LABEL.PY FIELDS

1.1.1 real_id

fields.Char : The ID of the label written on the label (8 characters)

1.1.2 type

fields.Char : The type of the label

1.1.3 status

fields.Char : The status of the label on imagotag server (online, offline, unregistered)

1.1.4 description

fields.Char : Nickname of the label given by the user

1.1.5 image

fields.Binary : Image that is loaded on page 0 of the label

1.1.6 status_color

fields.Integer : Color of the label in kanban view

1.1.7 is_created

fields.Boolean : Tag used to make impossible the change of real_id after the creation of the label

1.1.8 need_update

fields.Boolean : Flag used to update the label

1.1.9 png

fields.Binary : Additional image that will be sent to the label.

1.1.10 task_id

fields.Char : ID of the last update transaction

1.1.11 task_status

fields.Char : Status of the last update transaction

1.1.12 products

fields.Many2many : The products that are linked to the label

1.1.13 len_products

fields.Integer : Size of products field

1.1.14 template

fields.Many2one : The template that is linked to the label

1.1.15 preview

fields.Binary : The preview of the label when you modify the matching

2 LABEL.PY METHODS

2.1.1 _get_len_products()

Compute method to get the length of the products field.

2.1.2 _get_type_and_status()

Compute method to get the type and the status of the label from imagotag server.

2.1.3 _get_status_task()

Compute method to get the status of the last image task sent to the label.

2.1.4 _check_color()

Compute method to set color of the label frame on Kanban view.

2.1.5 _get_last_image_loaded()

Compute method that finds the task from last transaction created from update_esl() and set the image to the last image loaded in the label
This image may be not the one displayed because the page needs to be switched before it appears

2.1.6 _get_image_from_task(task_id)

Get the image sent in the task with the id task_id

2.1.7 _need_update()

Compute method to set the flag to True when one of the fields that is inside the xml sent is modified.

2.1.8 _update_esls(preload)

Method that calls _update_esl(preload) to the labels through imagotag server with the flag need_update set on True.

2.1.9 _update_esl(preload)

Method that update the label it is called from. And set the flag need_update to False.

2.1.10 _build_task_body(preload)

Method that build the xml's tasks content for both management template and linked template.

2.1.11 _xml_content(products, template, page, preload)

Builds the XML that will be sent to imagotag server

2.1.12 _get_preview()

Compute method to preview the image that will be sent to the labels once the update is done

2.1.13 _switch_page(page)

Method to switch page between inventory template and sale template

2.1.14 _register_label(Label_id)

Method to register the label with id: Label_id on the imagotag server

2.1.15 _unregister_label(Label_id)

Method to unregister the label with id: Label_id on the imagotag server

2.1.16 is_registered(label)

Checks if the label with id: label is registered on imagotag server

2.1.17 create(vals)

Override create method to add some restrictions on the model and register the label on imagotag server

2.1.18 write(vals)

Override write method to set preview of the template used in the label

2.1.19 unlink()

Override unlink method to unregister the label from imagotag server when it's deleted

2.1.20 force_update()

Action called when user clicks on Update ESL

2.1.21 form_to_matching()

Action/View transition called when user clicks on Create/Modify Matching

2.1.22 new_match()

Action called when user create a new match from one or more product.product

2.1.23 new_match_template()

Action called when user create a new match from one or more product.template

2.1.24 _check_multi_products()

Constraint on the number of products and the type of the template.

3 PRODUCT.PY PRODUCT

3.1.1 matching

fields.Many2many : Labels that are linked to this product.

3.1.2 label_price

fields.Float : Price of the product that will be sent to the label.

3.1.3 label_discount_percent

fields.Float : Discount in percent that will be sent to the label.

3.1.4 label_discount_fixed

fields.Float : Discount fixed in the price's currency that will be sent to the label.

3.1.5 len_matching

fields.Integer : Number of labels that are linked to this product.

3.1.6 get_pos_price()

Compute method to get the price and the discounts from the point of sale's pricelist linked to the module.

3.1.7 get_len_matching()

Compute method to get the number of labels that are linked to this product.

4 PRODUCT.PY TEMPLATE

4.1.1 matching

fields.Many2many : Labels that are linked to this product.

4.1.2 len_matching_template

fields.Integer : Number of labels that are linked to this product.

4.1.3 get_matching()

Compute method to get all the labels that are linked to all its product's variants.

4.1.4 get_len_matching_template()

Compute method to get the number of labels that are linked to all its product's variants.

4.1.5 action_view_matchings()

Action to display all the labels from matching

5 TEMPLATE.PY

5.1.1 name

fields.Char : Name of the template on imagotag server (with the .xsl extension)

5.1.2 description

fields.Char : Name of the template inside Odoo, will be selected when creating/modifying a matching

5.1.3 type

fields.Char : Type of the label(s) that is able to support this template

5.1.4 multi

fields.Boolean : Set to true if the template can handle multiple products

5.1.5 dyn

fields.Boolean : Set to true if the template dynamically adapts to the size of the label

5.1.6 preview

fields.Binary : Last imaged loaded in a label that uses this template

5.1.7 label

fields.One2many : Labels that are linked to this template

5.1.8 config

fields.One2many : Settings that use this template as a management template

5.1.9 _get_type_template()

Compute method to analyze the template file to get which type of label it supports

5.1.10 set_preview(image)

Method to set the preview of the template. Is called in write(vals) of Label class

6 RES_CONFIG.PY

6.1.1 core_appliance_ip

fields.Char : IP address of the imagotag server

6.1.2 pos

fields.Many2one : Point of Sale Configuration of the store that will be used to get prices

6.1.3 template_gestion

fields.Many2one : Template that will be used by all labels on page 1 (Management Template)

6.1.4 get_default_core_appliance_ip(fields)

Getter of the core_appliance_ip field

6.1.5 get_default_template_gestion(fields)

Getter of the template_gestion field

6.1.6 get_default_pos(fields)

Getter of the pos fields

6.1.7 set_default_core_appliance_ip()

Setter of the core_appliance_ip field

6.1.8 set_default_template_gestion()

Setter of the template_gestion field

6.1.9 set_default_pos()

Setter of the pos field