

Falinwa Technical Guide



Developer Tools

1. Sublime 3 / similar application that can check code

Sublime 3 package to install:

- Anaconda
- Trailing Space

VS Code extension to install:

- Black Fomatter

2. Github Desktop / Smartgit (optional)

3. PgAdmin (optional)

Development Rules

1. Creating field in odoo Model/Object

need to add “fal_” in front of field name.

example: fal_partner_id

```
class StockMoveLine(models.Model):  
    _inherit = "stock.move.line"  
  
    fal_partner_id = fields.Many2one('res.partner', string="Partner")
```

2. Create New Model/Object

need to add “fal.” in front of object name.

example: fal.new.model

and need to add _description in new model.

for new model we dont need to put “fal_” in front of field name.

```
class FalNewModel(models.Model):  
    _name = 'fal.new.model'  
    _description = 'New Model'  
  
    partner_id = fields.Many2one('res.partner', string="Partner")
```

Development Rules

3. Add comment in function

if there is complex function or long function need to add comment to make other developer easy to understand.

```
result[out] = (result[out][0] + taken_from_stock, False)

# Reconcile with the ins.
# The while loop will finish because it will pop from ins_per_product or decrease the demand until zero
if not float_is_zero(demand, precision_rounding=product_rounding):
    demand = _reconcile_out_with_ins(result, out, ins_per_product[product.id], demand, product_rounding, only_matching_move_dest=True)
if not float_is_zero(demand, precision_rounding=product_rounding):
    unreconciled_outs.append((demand, out))

for demand, out in unreconciled_outs:
    remaining = _reconcile_out_with_ins(result, out, ins_per_product[product.id], demand, product_rounding, only_matching_move_dest=False)
    if not float_is_zero(remaining, precision_rounding=out.product_id.uom_id.rounding) and out not in result:
        result[out] = (-remaining, False)
```

Development Rules

4. Override Odoo Function

When we override odoo function need to add comment “Falinwa Override”. and add “Falinwa start here” in the line that you modify and “Falinwa end here”.

purpose: when we migrate the module we know this is odoo function and need to check the funtion in new odoo version

notes: change the comment from falinwa to CLuedoo if it's CLuedoo module

```
# Cluedoo Override
@api.model
def format_value(self, amount, currency=False, blank_if_zero=False):
    currency_id = currency or self.env.company.currency_id
    # With IFRS, need to use group currency
    if self.env.context.get('ifrs') and self.env.company.group_currency_id:
        currency_id = self.env.company.group_currency_id
    # Change End Here
    if currency_id.is_zero(amount):
        if blank_if_zero:
            return ''
        # don't print -0.0 in reports
        amount = abs(amount)
    # Cluedoo End Here
    if self.env.context.get('no_format'):
        return amount
    return formatLang(self.env, amount, currency_obj=currency_id)
```

Development Rules

5. Super Odoo Function

When we super odoo function need to add comment “Falinwa Super”.

purpose: when we migrate the module we know this is odoo function and need to check the funtion in new odoo version

notes: change the comment from falinwa to CLuedoo if it's CLuedoo module

```
# Cluedoo Super
@api.model
def _prepare_sale_order_line_data(self, line, company):
    res = super(PurchaseOrder, self).prepare_sale_order_line_data(line, company)
    vals = []
    for item in line.product_custom_attribute_value_ids:
        vals.append((0, 0, {
            'name': item.name,
            'custom_product_template_attribute_value_id': item.custom_product_template_attribute_value_id.id,
            'custom_value': item.custom_value,
        }))
    res['product_no_variant_attribute_value_ids'] = [(6, 0, line.product_no_variant_attribute_value_ids.ids)]
    res['product_custom_attribute_value_ids'] = vals
    return res
```

Development Rules

6. Manifest

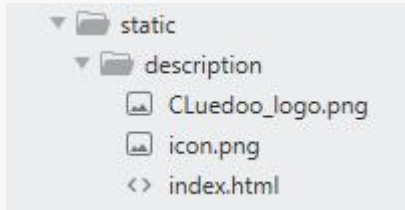
In Manifest file please make sure have a support, author, license, website and update version if needed.

```
{
  "name": "Product Configurator on Intercompany",
  "version": "15.0.1.0.0",
  "license": 'OPL-1',
  "summary": 'Parsing product attribute data',
  "category": 'Productivity',
  "author": 'CLuedoo',
  "website": "https://www.cluedoo.com",
  "support": 'cluedoo@falinwa.com',
  "description": ""
    1. Parsing product attribute data.
  "",
  "depends": [
    'sale_purchase_inter_company_rules',
    'fal_purchase_product_configurator',
  ],
  "data": [
  ],
  "demo": [
  ],
  "price": 180.00,
  "currency": 'EUR',
  "application": False,
}
```

Development Rules

7. Icon / CLuedoo Logo and index.html

Please add Icon / CLuedoo logo (for CLuedoo module).and index.html

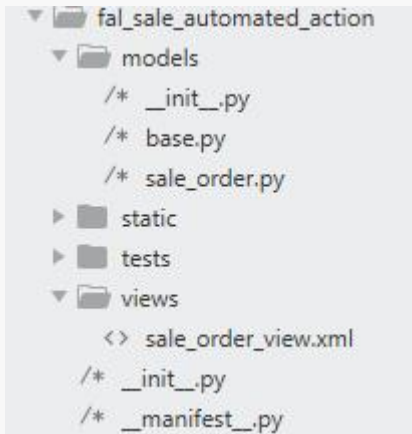


Development Rules

8. File Naming

the file name should be related to what we have inside the file.

example: we have model sale.order in the file. so the file name should be sale_order.py or sale_order_view.xml



Development Rules

9. ID naming

when we add id in views please make sure is not exist in other module. and add “fal_” in the front of id.

```
<record model="ir.ui.view" id="fal_vprocess_form_view">
  <field name="active" eval="True"/>
  <field name="field_parent" eval="False"/>
  <field name="groups_id" eval="[(6, 0, [])]"/>
  <field name="inherit_id" eval="False"/>
```

10. Push Module

before push module, please make sure we already remove the log/print to make a code looks clean.