Technical Guidelines for Common Errors in Odoo 18 App Development

Table of Contents

Module Structure and Manifest Issues {#module-structure}

1.1 Missing or Incorrect __manifest__.py

Error: Module not recognized by Odoo

python

```
# ❌ WRONG - Missing required fields
{
    'name': 'My Module',
}


# ✅ CORRECT - Complete manifest
{
    'name': 'Real Estate Management',
    'version': '18.0.1.0.0',
    'summary': 'Manage real estate properties and listings',
    'description': """
    Complete real estate management system with:
    - Property listings
    - Customer management
    - Sales tracking
    """,
    'author': 'Your Company',
```

```
    'website': 'https://www.yourcompany.com',
    'category': 'Sales',
    'depends': ['base', 'mail'],
    'data': [
        'security/ir.model.access.csv',
        'views/estate_property_views.xml',
        'views/estate_menus.xml',
    ],
    'installable': True,
    'application': True,
    'license': 'LGPL-3',
}
```

1.2 Incorrect Module Directory Structure

Error: ImportError or module components not loading

```
# ❌ WRONG Structure
my_module/
 ├── __manifest__.py
 ├── models.py  # Should be in models/ directory
 └── views.xml  # Should be in views/ directory

# ✅ CORRECT Structure
my_module/
 ├── __init__.py
 ├── __manifest__.py
 ├── models/
 |   ├── __init__.py
 |   └── estate_property.py
 ├── views/
 |   ├── estate_property_views.xml
 |   └── estate_menus.xml
```

```
└── security/
    └── ir.model.access.csv
```

1.3 Missing __init__.py Files

Error: Python modules not imported

python

```python
# ❌ MISSING - No __init__.py files

# ✅ CORRECT - Main module __init__.py
from . import models

# ✅ CORRECT - models/__init__.py
from . import estate_property
```

Model Definition Errors {#model-errors}

2.1 Missing Model Registration

Error: AttributeError: Model not found

python

```python
# ❌ WRONG - Missing _name attribute
class EstateProperty(models.Model):
    # Missing _name = 'estate.property'
    name = fields.Char(string="Title")

# ✅ CORRECT - Proper model definition
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'
    _order = 'name asc'

    name = fields.Char(string="Title", required=True)
```

2.2 Field Definition Errors

Error: ValueError: Invalid field definition

python

```python
# ❌ WRONG - Incorrect field syntax
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: missing fields module
    name = Char(string="Title")
    # Wrong: incorrect Many2one syntax
    property_type = fields.Many2one('estate.property.type')

# ✅ CORRECT - Proper field definitions
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    name = fields.Char(string="Title", required=True)
    property_type_id = fields.Many2one(
        'estate.property.type',
        string="Property Type"
    )
    expected_price = fields.Float(
        string="Expected Price",
        digits=(16, 2),
        required=True
    )
```

2.3 Computed Field Errors

Error: RecursionError or computed field not updating

python

```python
# ❌ WRONG - Missing @api.depends or incorrect computation
class EstateProperty(models.Model):
    _name = 'estate.property'
```

```
    total_area = fields.Integer(compute='_compute_total_area')

    def _compute_total_area(self):  # Missing @api.depends
        self.total_area = self.living_area + self.garden_area  # No loop

# ✅ CORRECT - Proper computed field
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    living_area = fields.Integer(string="Living Area (sqm)")
    garden_area = fields.Integer(string="Garden Area (sqm)")
    total_area = fields.Integer(
        string="Total Area (sqm)",
        compute='_compute_total_area',
        store=True
    )

    @api.depends('living_area', 'garden_area')
    def _compute_total_area(self):
        for record in self:
            record.total_area = record.living_area + record.garden_area
```

View and UI Issues {#view-errors}

3.1 XML Syntax Errors

Error: ParseError: XML syntax error

xml

```
<!-- ❌ WRONG - Missing closing tags, incorrect structure -->
<odoo>
<data>
```

```
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form>
      <field name="name">  <!-- Missing closing tag -->
    </form>
  </field>
<!-- Missing closing record tag -->
</data>
</odoo>

<!-- ✅ CORRECT - Proper XML structure -->
<odoo>
  <data>
    <record id="view_estate_property_form" model="ir.ui.view">
      <field name="name">estate.property.form</field>
      <field name="model">estate.property</field>
      <field name="arch" type="xml">
        <form string="Property Details">
          <sheet>
            <group>
              <field name="name"/>
              <field name="expected_price"/>
            </group>
          </sheet>
        </form>
      </field>
    </record>
  </data>
</odoo>
```

3.2 Field Not Found in View

Error: ValueError: Field 'field_name' does not exist

xml

```xml
<!-- ❌ WRONG - Field doesn't exist in model -->
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form>
      <field name="non_existent_field"/>  <!-- Field not defined in model -->
    </form>
  </field>
</record>

<!-- ✅ CORRECT - Using existing model fields -->
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form string="Property Details">
      <sheet>
        <group>
          <field name="name"/>  <!-- Field exists in model -->
          <field name="expected_price"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

3.3 Action Configuration Errors

Error: Action not working or views not loading

```xml
<!-- ❌ WRONG - Missing or incorrect action configuration -->
<record id="action_estate_property" model="ir.actions.act_window">
  <field name="name">Properties</field>
  <!-- Missing res_model -->
  <field name="view_mode">tree,form</field>
</record>

<!-- ✅ CORRECT - Complete action definition -->
<record id="action_estate_property" model="ir.actions.act_window">
  <field name="name">Properties</field>
  <field name="res_model">estate.property</field>
  <field name="view_mode">tree,form,kanban</field>
  <field name="search_view_id" ref="view_estate_property_search"/>
  <field name="help" type="html">
    <p class="o_view_nocontent_smiling_face">
      Create your first property listing!
    </p>
  </field>
</record>
```

Security and Access Rights Problems {#security-errors}

4.1 Missing Access Rights

Error: AccessError: You are not allowed to access this document

```csv
# ❌ WRONG - Missing ir.model.access.csv or incorrect entries
# File: security/ir.model.access.csv (empty or missing)

# ✅ CORRECT - Proper access rights
# File: security/ir.model.access.csv
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
```

access_estate_property_user,estate.property.user,model_estate_property,base.group_user,1,1,1,1

access_estate_property_type_user,estate.property.type.user,model_estate_property_type,base.group_user,1,1,1,1

4.2 Security File Not Listed in Manifest

Error: Access rights not loaded

python

```python
# ❌ WRONG - Security file not in manifest data
{
    'name': 'Estate Module',
    'depends': ['base'],
    'data': [
        'views/estate_property_views.xml',
        # Missing 'security/ir.model.access.csv',
    ],
}

# ✅ CORRECT - Security file included
{
    'name': 'Estate Module',
    'depends': ['base'],
    'data': [
        'security/ir.model.access.csv',  # Security first
        'views/estate_property_views.xml',
        'views/estate_menus.xml',
    ],
}
```

Field and Relationship Errors {#field-errors}

5.1 Many2one Field Errors

Error: ValueError: Invalid Many2one reference

python

```python
# ❌ WRONG - Incorrect Many2one definition
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: target model doesn't exist
    property_type_id = fields.Many2one('non.existent.model')
    # Wrong: missing comodel_name parameter name
    user_id = fields.Many2one('res.users', required=True)

# ✅ CORRECT - Proper Many2one fields
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    property_type_id = fields.Many2one(
        'estate.property.type',
        string="Property Type"
    )
    user_id = fields.Many2one(
        'res.users',
        string="Salesperson",
        default=lambda self: self.env.user
    )
```

5.2 One2many Relationship Errors

Error: ValueError: Wrong One2many inverse field

python

```python
# ❌ WRONG - Incorrect One2many setup
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: inverse field doesn't exist in comodel
```

```
    offer_ids = fields.One2many('estate.property.offer', 'wrong_field')

class EstatePropertyOffer(models.Model):
    _name = 'estate.property.offer'
    # Missing the inverse Many2one field

# ✅ CORRECT - Proper One2many relationship
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    offer_ids = fields.One2many(
        'estate.property.offer',
        'property_id',  # This field must exist in estate.property.offer
        string="Offers"
    )

class EstatePropertyOffer(models.Model):
    _name = 'estate.property.offer'
    _description = 'Property Offer'

    property_id = fields.Many2one(
        'estate.property',
        string="Property",
        required=True,
        ondelete='cascade'
    )
```

Server and Installation Issues {#server-errors}

6.1 Module Update Errors

Error: Module not updating after changes

bash

```
# ❌ WRONG - Not updating module after changes
./odoo-bin --addons-path=addons -d mydb

# ✅ CORRECT - Update module after changes
./odoo-bin --addons-path=addons -d mydb -u estate_module

# ✅ For development - auto-reload
./odoo-bin --addons-path=addons -d mydb --dev xml,reload
```

6.2 Import Errors

Error: ImportError: cannot import name

python

```
# ❌ WRONG - Incorrect import statements
from openerp import models, fields, api  # Wrong for Odoo 18
from odoo import field  # Wrong module name

# ✅ CORRECT - Proper imports for Odoo 18
from odoo import models, fields, api
from odoo.exceptions import ValidationError, UserError
```

6.3 Database Connection Issues

Error: psycopg2.OperationalError: database connection failed

bash

```
# ❌ Common issues:
# - PostgreSQL not running
# - Wrong database credentials
# - User doesn't have permissions

# ✅ SOLUTIONS:
# Check PostgreSQL status
sudo systemctl status postgresql
```

```
# Create Odoo user in PostgreSQL
sudo -u postgres createuser -s $USER

# Test database connection
psql -h localhost -U $USER -d postgres
```

Best Practices and Prevention {#best-practices}

7.1 Development Workflow

1. **Always use version control (Git)**

```bash
git init
git add .
git commit -m "[ADD] Initial module structure"
```

2. **Test incrementally**

```bash
# After each major change
./odoo-bin --addons-path=addons -d test_db -u your_module --test-enable
```

3. **Use developer mode**

```
# Add to URL for debugging
http://localhost:8069/web?debug=1
```

7.2 Code Quality Checklist

```python
# ✅ Model Definition Checklist
class MyModel(models.Model):
    _name = 'my.model'                # ✓Required
    _description = 'My Model'         # ✓Good practice
    _order = 'name asc'               # ✓Default sorting

    # ✓Field definitions with proper attributes
    name = fields.Char(string="Name", required=True)
```

```
# ✓Computed fields with dependencies
@api.depends('field1', 'field2')
def _compute_something(self):
    for record in self:
        record.computed_field = record.field1 + record.field2

# ✓Constraints with clear error messages
@api.constrains('price')
def _check_price_positive(self):
    for record in self:
        if record.price <= 0:
            raise ValidationError("Price must be positive!")
```

7.3 Common Prevention Strategies

1. **Always backup before major changes**
2. **Use meaningful names for models, fields, and methods**
3. **Follow Odoo naming conventions**
4. **Test on a copy database first**
5. **Check server logs regularly**
6. **Use proper field types and attributes**
7. **Define access rights from the beginning**
8. **Document your code**

7.4 Debugging Tools

python

```
# Add debugging breakpoints
import pdb; pdb.set_trace()

# Log important information
import logging
_logger = logging.getLogger(__name__)
_logger.info("Debug message: %s", variable)
```

```
# Use Odoo shell for testing
./odoo-bin shell -d database_name
```

7.5 Performance Considerations

python

```
# ✅ Efficient database queries
# Use proper field indexing
name = fields.Char(string="Name", index=True)

# ✅ Batch operations
for record in self:
    # Process records efficiently

# ✅ Avoid N+1 queries
records = self.env['model.name'].search([])
records.mapped('related_field.name')  # Better than loop
```

Quick Reference Commands

bash

```
# Module operations
./odoo-bin -d mydb -u module_name          # Update module
./odoo-bin -d mydb -i module_name          # Install module
./odoo-bin -d mydb --dev xml,reload        # Development mode

# Database operations
createdb mydb                              # Create database
dropdb mydb                                # Delete database

# Debugging
./odoo-bin shell -d mydb                   # Odoo shell
./odoo-bin --log-level=debug               # Debug logging
```

This guide covers the most common errors and issues encountered in Odoo 18 app development. Always refer to the official Odoo 18.0 documentation for the most up-to-date information and detailed explanations.

Top 50 Common Errors in Odoo 18 App Development - Detailed Guide

Table of Contents

Module Structure and Manifest Issues (Errors 1-10) {#module-structure}

Error 1: Missing __manifest__.py File

Symptoms: Module not appearing in Apps list, "No module named" errors

Problem:

```
my_module/
├── models/
│   └── estate_property.py
└── views/
    └── estate_property_views.xml
# Missing __manifest__.py
```

Solution:

python

```
# my_module/__manifest__.py
{
    'name': 'Real Estate Management',
    'version': '18.0.1.0.0',
    'summary': 'Manage real estate properties efficiently',
    'description': """
Real Estate Management System
============================
Complete solution for managing:
* Property listings
```

```
* Customer relationships
* Sales processes
* Reporting and analytics
    """,
    'author': 'Your Company Name',
    'website': 'https://www.yourcompany.com',
    'category': 'Sales',
    'depends': ['base', 'mail'],
    'data': [
        'security/ir.model.access.csv',
        'views/estate_property_views.xml',
        'views/estate_menus.xml',
        'data/estate_property_data.xml',
    ],
    'demo': [
        'demo/estate_property_demo.xml',
    ],
    'assets': {
        'web.assets_backend': [
            'my_module/static/src/css/estate.css',
            'my_module/static/src/js/estate.js',
        ],
    },
    'installable': True,
    'application': True,
    'auto_install': False,
    'license': 'LGPL-3',
}
```

Error 2: Incorrect Module Name Convention

Symptoms: Module installation fails, naming conflicts

Problem:

python

```python
# ❌ WRONG - Spaces, capitals, special characters
{
    'name': 'My Real-Estate Module!',
    # Directory name: My Real-Estate Module!
}
```

Solution:

python

```python
# ✅ CORRECT - Use underscores, lowercase
# Directory name: real_estate_management
{
    'name': 'Real Estate Management',  # Display name can have spaces
    'technical_name': 'real_estate_management',  # Matches directory
}
```

Error 3: Missing or Incorrect __init__.py Files

Symptoms: ImportError, models not loading, "No module named" errors

Problem:

```
real_estate_management/
 ├── __manifest__.py
 ├── models/          # Missing __init__.py
 |   └── estate_property.py
 └── views/
     └── estate_views.xml
```

Solution:

python

```python
# real_estate_management/__init__.py
from . import models
from . import controllers  # If you have controllers
from . import wizards      # If you have wizards

# real_estate_management/models/__init__.py
```

```
from . import estate_property
from . import estate_property_type
from . import estate_property_offer
```

Error 4: Circular Dependencies in Manifest

Symptoms: Module installation fails, dependency resolution errors

Problem:

python

```
# Module A depends on Module B
# Module B depends on Module A
# real_estate/__manifest__.py
{
    'depends': ['base', 'property_management'],
}

# property_management/__manifest__.py
{
    'depends': ['base', 'real_estate'],  # Circular dependency
}
```

Solution:

python

```
# ✅ Create a base module that both depend on
# real_estate_base/__manifest__.py
{
    'name': 'Real Estate Base',
    'depends': ['base'],
}

# real_estate/__manifest__.py
{
    'depends': ['real_estate_base'],
}
```

```python
# property_management/__manifest__.py
{
    'depends': ['real_estate_base'],
}
```

Error 5: Incorrect File Paths in Manifest Data

Symptoms: XML files not loading, "File not found" errors

Problem:

python

```python
# ❌ WRONG - Incorrect paths
{
    'data': [
        'security/access.csv',          # File is ir.model.access.csv
        'views/property_views.xml',     # File is estate_property_views.xml
        'data/demo_data.xml',           # File is in demo/ folder
    ],
}
```

Solution:

python

```python
# ✅ CORRECT - Exact file paths
{
    'data': [
        'security/ir.model.access.csv',
        'views/estate_property_views.xml',
        'views/estate_property_type_views.xml',
        'views/estate_menus.xml',
        'data/estate_property_data.xml',
    ],
    'demo': [
        'demo/estate_property_demo.xml',
    ],
```

}

Error 6: Missing Dependencies in Manifest

Symptoms: NameError, AttributeError when accessing other modules' features

Problem:

python

```
# ❌ Using features without declaring dependencies
{
    'name': 'Real Estate',
    'depends': ['base'],  # Missing 'mail', 'website', etc.
}

# In your model:
class EstateProperty(models.Model):
    _inherit = ['mail.thread']  # Error: mail module not in depends
```

Solution:

python

```
# ✅ CORRECT - Include all required dependencies
{
    'name': 'Real Estate',
    'depends': [
        'base',
        'mail',         # For mail.thread inheritance
        'website',      # If using website features
        'sale',         # If extending sale functionality
    ],
}
```

Error 7: Incorrect Version Format

Symptoms: Version comparison issues, upgrade problems

Problem:

python

```
# ❌ WRONG - Incorrect version formats
```

```
{
    'version': '1.0',          # Too simple
    'version': '18.0.1',       # Missing patch version
    'version': 'v1.0.0',       # Invalid prefix
}
```

Solution:

python

```
# ✅ CORRECT - Semantic versioning for Odoo 18
{
    'version': '18.0.1.0.0',    # odoo.series.major.minor.patch
    # 18.0 = Odoo version
    # 1 = Major module version
    # 0 = Minor features
    # 0 = Patch/bugfix
}
```

Error 8: Incorrect Category Values

Symptoms: Module appears in wrong section, poor organization

Problem:

python

```
# ❌ WRONG - Non-standard or incorrect categories
{
    'category': 'My Category',          # Non-standard
    'category': 'Real Estate',          # Too specific
    'category': 'Custom/Real Estate',   # Invalid format
}
```

Solution:

python

```
# ✅ CORRECT - Use standard Odoo categories
{
    'category': 'Sales',              # Standard category
    # Other valid categories:
```

```
    # 'Accounting', 'Productivity', 'Website',
    # 'Manufacturing', 'Human Resources', 'Marketing',
    # 'Point of Sale', 'Purchases', 'Project', 'Inventory'
}
```

Error 9: Missing Application Flag for Main Modules

Symptoms: Module doesn't appear as main app, poor UX

Problem:

python

```
# ❌ WRONG - Missing application flag for main module
{
    'name': 'Real Estate Management',
    'installable': True,
    # Missing 'application': True
}
```

Solution:

python

```
# ✅ CORRECT - Set application flag for main modules
{
    'name': 'Real Estate Management',
    'application': True,       # Shows as main app
    'installable': True,
    'auto_install': False,     # Don't auto-install

    # For sub-modules or extensions:
    # 'application': False,    # Not a main app
    # 'auto_install': True,    # Auto-install with dependencies
}
```

Error 10: Incorrect License Specification

Symptoms: Legal compliance issues, app store rejection

Problem:

python

```
# ❌ WRONG - Incorrect or missing license
{
    'license': 'GPL',          # Too generic
    'license': 'MIT',          # Not compatible with Odoo
    # Missing license field
}
```

Solution:

python

```
# ✅ CORRECT - Use appropriate license for Odoo
{
    'license': 'LGPL-3',       # For community modules
    # or
    'license': 'OEEL-1',       # For enterprise modules
    # or
    'license': 'Other proprietary', # For custom modules
}
```

Model Definition Errors (Errors 11-20) {#model-errors}

Error 11: Missing _name Attribute

Symptoms: AttributeError: Model class has no _name, registration fails

Problem:

python

```
# ❌ WRONG - Missing _name attribute
class EstateProperty(models.Model):
    # Missing _name = 'estate.property'

    name = fields.Char(string="Title")
    price = fields.Float(string="Price")
```

Solution:

python

```
# ✅ CORRECT - Always include _name for new models
```

```
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'
    _order = 'name asc'
    _rec_name = 'name'  # Field to use for record representation

    name = fields.Char(string="Title", required=True)
    price = fields.Float(string="Price", digits=(16, 2))
```

Error 12: Incorrect Model Inheritance

Symptoms: TypeError, inheritance not working, duplicate _name

Problem:

python

```
# ❌ WRONG - Incorrect inheritance syntax
class EstateProperty(models.Model):
    _name = 'estate.property'
    _inherit = 'estate.property'  # Same as _name - creates recursion

# ❌ WRONG - Wrong way to extend existing model
class ResPartner(models.Model):
    _name = 'res.partner.extended'  # Creates new model instead of extending
    _inherit = 'res.partner'
```

Solution:

python

```
# ✅ CORRECT - Extending existing model (prototypal inheritance)
class ResPartner(models.Model):
    _inherit = 'res.partner'  # No _name when extending

    is_real_estate_agent = fields.Boolean(string="Is Real Estate Agent")
    property_ids = fields.One2many('estate.property', 'agent_id', string="Properties")

# ✅ CORRECT - Creating new model with mixins
```

```
class EstateProperty(models.Model):
    _name = 'estate.property'
    _inherit = ['mail.thread', 'mail.activity.mixin']  # Inherit from mixins
    _description = 'Real Estate Property'

    name = fields.Char(string="Title", required=True, tracking=True)
```

Error 13: Wrong Field Type Selection

Symptoms: Data type errors, validation issues, poor UX

Problem:

python

```
# ❌ WRONG - Inappropriate field types
class EstateProperty(models.Model):
    _name = 'estate.property'

    price = fields.Char(string="Price")            # Should be Float
    is_available = fields.Char(string="Available")   # Should be Boolean
    bedrooms = fields.Float(string="Bedrooms")       # Should be Integer
    description = fields.Char(string="Description")  # Should be Text
```

Solution:

python

```
# ✅ CORRECT - Appropriate field types
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    price = fields.Float(
        string="Price",
        digits=(16, 2),          # Precision for currency
        required=True
    )
    is_available = fields.Boolean(
```

```
        string="Available",
        default=True
    )
    bedrooms = fields.Integer(
        string="Bedrooms",
        default=1
    )
    description = fields.Text(
        string="Description",
        help="Detailed property description"
    )
    state = fields.Selection([
        ('draft', 'Draft'),
        ('published', 'Published'),
        ('sold', 'Sold'),
        ('canceled', 'Canceled')
    ], string="Status", default='draft')
```

Error 14: Missing Field Constraints and Validation

Symptoms: Invalid data in database, business rule violations

Problem:

python

```
# ❌ WRONG - No validation on critical fields
class EstateProperty(models.Model):
    _name = 'estate.property'

    price = fields.Float(string="Price")       # No validation
    bedrooms = fields.Integer(string="Bedrooms") # Could be negative
    email = fields.Char(string="Email")        # No email validation
```

Solution:

python

```
# ✅ CORRECT - Proper validation and constraints
```

```
from odoo.exceptions import ValidationError
import re

class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    price = fields.Float(
        string="Price",
        required=True,
        digits=(16, 2)
    )
    bedrooms = fields.Integer(
        string="Bedrooms",
        default=1
    )
    email = fields.Char(string="Contact Email")

    # SQL constraints
    _sql_constraints = [
        ('price_positive', 'CHECK(price > 0)', 'Price must be positive!'),
        ('bedrooms_positive', 'CHECK(bedrooms >= 0)', 'Bedrooms cannot be negative!'),
    ]

    # Python constraints
    @api.constrains('email')
    def _check_email_format(self):
        for record in self:
            if record.email and not re.match(r'^[^@]+@[^@]+\.[^@]+$', record.email):
                raise ValidationError("Please enter a valid email address!")
```

```
    @api.constrains('price', 'bedrooms')
    def _check_price_per_bedroom(self):
        for record in self:
            if record.bedrooms > 0 and record.price / record.bedrooms < 1000:
                raise ValidationError("Price per bedroom seems too low!")
```

Error 15: Incorrect Computed Field Implementation

Symptoms: RecursionError, computed fields not updating, performance issues

Problem:

python

```
# ❌ WRONG - Missing dependencies, wrong computation logic
class EstateProperty(models.Model):
    _name = 'estate.property'

    living_area = fields.Integer(string="Living Area")
    garden_area = fields.Integer(string="Garden Area")
    total_area = fields.Integer(compute='_compute_total_area')

    def _compute_total_area(self):  # Missing @api.depends
        self.total_area = self.living_area + self.garden_area  # No loop for recordset
```

Solution:

python

```
# ✅ CORRECT - Proper computed field with dependencies
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    living_area = fields.Integer(string="Living Area (sqm)")
    garden_area = fields.Integer(string="Garden Area (sqm)")
    total_area = fields.Integer(
        string="Total Area (sqm)",
        compute='_compute_total_area',
```

```
    store=True,  # Store for performance if needed for search/group
    help="Total area including living area and garden"
)

@api.depends('living_area', 'garden_area')
def _compute_total_area(self):
    for record in self:
        record.total_area = (record.living_area or 0) + (record.garden_area or 0)

# For inverse computation (making computed field writable)
def _inverse_total_area(self):
    for record in self:
        if record.garden_area:
            record.living_area = record.total_area - record.garden_area
        else:
            record.living_area = record.total_area
```

Error 16: Improper Use of @api Decorators

Symptoms: TypeError, methods not working as expected, wrong context

Problem:

python

```
# ❌ WRONG - Incorrect @api decorator usage
class EstateProperty(models.Model):
    _name = 'estate.property'

    @api.one  # Deprecated in Odoo 18
    def get_display_name(self):
        return self.name

    @api.multi  # Wrong decorator for model methods
    def create_property(self, vals):
        return super().create(vals)
```

Solution:

python

```python
# ✅ CORRECT - Proper @api decorator usage in Odoo 18
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    @api.depends('name', 'property_type_id')
    def _compute_display_name(self):
        for record in self:
            if record.property_type_id:
                record.display_name = f"{record.name} ({record.property_type_id.name})"
            else:
                record.display_name = record.name

    @api.model
    def create(self, vals):
        # @api.model for methods that don't operate on recordsets
        return super().create(vals)

    def write(self, vals):
        # No decorator needed for standard record methods
        return super().write(vals)

    @api.onchange('property_type_id')
    def _onchange_property_type(self):
        # @api.onchange for UI reactive methods
        if self.property_type_id:
            self.expected_price = self.property_type_id.default_price
```

Error 17: Incorrect Default Value Implementation

Symptoms: Default values not working, errors on record creation

Problem:

python

```python
# ❌ WRONG - Incorrect default value syntax
class EstateProperty(models.Model):
    _name = 'estate.property'

    create_date = fields.Datetime(default=datetime.now())  # Evaluated once at module load
    user_id = fields.Many2one('res.users', default=self.env.user)  # Wrong context
    sequence = fields.Integer(default=self._get_next_sequence())  # Method doesn't exist yet
```

Solution:

python

```python
# ✅ CORRECT - Proper default value implementation
from datetime import datetime, timedelta

class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    create_date = fields.Datetime(
        default=fields.Datetime.now  # Method reference, not call
    )
    user_id = fields.Many2one(
        'res.users',
        string="Responsible User",
        default=lambda self: self.env.user  # Lambda function
    )
    available_from = fields.Date(
        string="Available From",
        default=lambda self: fields.Date.today() + timedelta(days=30)
    )
    reference = fields.Char(
```

```
        string="Reference",
        default=lambda self: self._generate_reference()
    )

    @api.model
    def _generate_reference(self):
        """Generate unique reference for property"""
        sequence = self.env['ir.sequence'].next_by_code('estate.property') or 'EST0001'
        return sequence
```

Error 18: Wrong Use of _rec_name and name_get

Symptoms: Poor record display in dropdowns, search issues

Problem:

python

```
# ❌ WRONG - Incorrect record name implementation
class EstateProperty(models.Model):
    _name = 'estate.property'

    title = fields.Char(string="Title")  # No 'name' field
    # _rec_name not set, no name_get method
```

Solution:

python

```
# ✅ CORRECT - Proper record naming implementation
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'
    _rec_name = 'title'  # Use 'title' field for record representation

    title = fields.Char(string="Title", required=True)
    property_type_id = fields.Many2one('estate.property.type', string="Type")
    reference = fields.Char(string="Reference")
```

```
    def name_get(self):
        """Custom name display for records"""
        result = []
        for record in self:
            if record.reference and record.property_type_id:
                name = f"[{record.reference}] {record.title} ({record.property_type_id.name})"
            elif record.reference:
                name = f"[{record.reference}] {record.title}"
            else:
                name = record.title
            result.append((record.id, name))
        return result

    @api.model
    def _name_search(self, name, args=None, operator='ilike', limit=100, name_get_uid=None):
        """Custom search logic for record names"""
        args = args or []
        if name:
            domain = ['|', '|',
                      ('title', operator, name),
                      ('reference', operator, name),
                      ('property_type_id.name', operator, name)]
            records = self.search(domain + args, limit=limit)
            return records.name_get()
        return super()._name_search(name, args, operator, limit, name_get_uid)
```

Error 19: Incorrect Model Ordering and Indexing

Symptoms: Poor performance, unexpected sort order

Problem:

python

```
# ❌ WRONG - No ordering specified, missing indexes on search fields
class EstateProperty(models.Model):
```

```
    _name = 'estate.property'

    name = fields.Char(string="Title")
    price = fields.Float(string="Price")
    create_date = fields.Datetime(string="Created")
    # No _order specified, no indexes on frequently searched fields
```

Solution:

python

```
# ✅ CORRECT - Proper ordering and indexing
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'
    _order = 'sequence asc, create_date desc, name asc'  # Multiple field ordering

    sequence = fields.Integer(string="Sequence", default=10, index=True)
    name = fields.Char(string="Title", required=True, index=True)  # Indexed for search
    price = fields.Float(string="Price", digits=(16, 2), index=True)  # Indexed for filtering
    state = fields.Selection([
        ('draft', 'Draft'),
        ('published', 'Published'),
        ('sold', 'Sold')
    ], string="Status", default='draft', index=True)  # Indexed for filtering

    create_date = fields.Datetime(string="Created", index=True)  # Indexed for sorting
    property_type_id = fields.Many2one(
        'estate.property.type',
        string="Type",
        index=True  # Foreign keys are automatically indexed, but explicit is clear
    )
```

Error 20: Missing or Incorrect Model Methods Override

Symptoms: Custom logic not executing, data inconsistency

Problem:

python

```python
# ❌ WRONG - Incorrect method override, missing super() calls
class EstateProperty(models.Model):
    _name = 'estate.property'

    def create(self, vals):
        # Custom logic here
        record = self.env['estate.property'].create(vals)  # Recursion!
        return record

    def write(self, vals):
        # Custom logic
        return True  # Not calling super()
```

Solution:

python

```python
# ✅ CORRECT - Proper method override with super() calls
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    @api.model
    def create(self, vals):
        # Pre-creation logic
        if not vals.get('reference'):
            vals['reference'] = self._generate_reference()

        # Call parent method
        record = super().create(vals)

        # Post-creation logic
```

```
    if record.agent_id:
        record._notify_agent_new_property()

    return record

def write(self, vals):
    # Pre-write logic
    if 'state' in vals and vals['state'] == 'sold':
        vals['sold_date'] = fields.Datetime.now()

    # Call parent method
    result = super().write(vals)

    # Post-write logic
    if 'price' in vals:
        self._log_price_change(vals['price'])

    return result

def unlink(self):
    # Check if records can be deleted
    if any(record.state == 'sold' for record in self):
        raise UserError("Cannot delete sold properties!")

    return super().unlink()

@api.model
def _generate_reference(self):
    return self.env['ir.sequence'].next_by_code('estate.property.reference')

def _notify_agent_new_property(self):
```

```
        # Send notification to agent
        template = self.env.ref('real_estate.email_template_new_property')
        template.send_mail(self.id)

    def _log_price_change(self, new_price):
        # Log price changes
        self.env['estate.property.log'].create({
            'property_id': self.id,
            'action': 'price_change',
            'new_value': new_price,
            'user_id': self.env.user.id,
        })
```

Field and Relationship Errors (Errors 21-30) {#field-errors}

Error 21: Incorrect Many2one Field Definition

Symptoms: ValueError, field not working, dropdown not populated

Problem:

python

```
# ❌ WRONG - Various Many2one definition issues
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: Missing comodel_name or incorrect syntax
    agent_id = fields.Many2one(string="Agent")  # No target model
    type_id = fields.Many2one('nonexistent.model', string="Type")  # Model doesn't exist
    user_id = fields.Many2one('res.users', required=True, default=1)  # Hard-coded ID
```

Solution:

python

```
# ✅ CORRECT - Proper Many2one field definitions
class EstateProperty(models.Model):
    _name = 'estate.property'
```

```
_description = 'Real Estate Property'

agent_id = fields.Many2one(
    'res.partner',
    string="Agent",
    domain="[('is_company', '=', False), ('category_id.name', '=', 'Real Estate Agent')]",
    help="Real estate agent responsible for this property"
)

property_type_id = fields.Many2one(
    'estate.property.type',
    string="Property Type",
    required=True,
    ondelete='restrict'  # Prevent deletion if properties exist
)

user_id = fields.Many2one(
    'res.users',
    string="Responsible User",
    required=True,
    default=lambda self: self.env.user,  # Dynamic default
    ondelete='cascade'
)

company_id = fields.Many2one(
    'res.company',
    string="Company",
    default=lambda self: self.env.company,
    index=True
)
```

```python
    # With context for creating related records
    buyer_id = fields.Many2one(
        'res.partner',
        string="Buyer",
        context="{'default_is_company': False, 'default_customer_rank': 1}"
    )
```

Error 22: Wrong One2many Inverse Field Setup

Symptoms: ValueError about inverse field, relationship not working

Problem:

python

```python
# ❌ WRONG - Incorrect One2many setup
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: inverse field doesn't exist in target model
    offer_ids = fields.One2many('estate.property.offer', 'wrong_field_name')

class EstatePropertyOffer(models.Model):
    _name = 'estate.property.offer'
    # Missing the inverse Many2one field 'property_id'

    price = fields.Float(string="Offer Price")
```

Solution:

python

```python
# ✅ CORRECT - Proper One2many relationship setup
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    offer_ids = fields.One2many(
        'estate.property.offer',
```

```
        'property_id',  # This field MUST exist in estate.property.offer
        string="Property Offers",
        copy=False  # Don't copy offers when duplicating property
    )

    # Computed field for offer count
    offer_count = fields.Integer(
        string="Number of Offers",
        compute='_compute_offer_count'
    )

    @api.depends('offer_ids')
    def _compute_offer_count(self):
        for record in self:
            record.offer_count = len(record.offer_ids)

class EstatePropertyOffer(models.Model):
    _name = 'estate.property.offer'
    _description = 'Property Offer'
    _order = 'price desc'  # Show highest offers first

    property_id = fields.Many2one(
        'estate.property',
        string="Property",
        required=True,
        ondelete='cascade'  # Delete offers when property is deleted
    )

    price = fields.Float(
        string="Offer Price",
        required=True,
```

```
        digits=(16, 2)
    )

    partner_id = fields.Many2one(
        'res.partner',
        string="Buyer",
        required=True
    )
```

Error 23: Incorrect Many2many Field Configuration

Symptoms: Relation table errors, performance issues, data corruption

Problem:

python

```
# ❌ WRONG - Incorrect Many2many configuration
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: No relation table specified, could conflict
    tag_ids = fields.Many2many('estate.property.tag')

    # Wrong: Incorrect column names
    feature_ids = fields.Many2many(
        'estate.property.feature',
        relation='estate_property_feature_rel',
        column1='wrong_col',  # Should match current model
        column2='wrong_col2'  # Should match target model
    )
```

Solution:

python

```
# ✅ CORRECT - Proper Many2many configuration
class EstateProperty(models.Model):
    _name = 'estate.property'
```

```
    _description = 'Real Estate Property'

    tag_ids = fields.Many2many(
        'estate.property.tag',
        relation='estate_property_tag_rel',    # Explicit relation table
        column1='property_id',                 # Column for this model
        column2='tag_id',                      # Column for target model
        string="Property Tags"
    )

    feature_ids = fields.Many2many(
        'estate.property.feature',
        relation='estate_property_feature_rel',
        column1='property_id',
        column2='feature_id',
        string="Property Features",
        domain="[('active', '=', True)]"       # Only active features
    )

    # With copy=False for certain relationships
    viewer_ids = fields.Many2many(
        'res.partner',
        relation='estate_property_viewer_rel',
        column1='property_id',
        column2='partner_id',
        string="Property Viewers",
        copy=False  # Don't copy viewers when duplicating
    )

class EstatePropertyTag(models.Model):
    _name = 'estate.property.tag'
```

```
    _description = 'Property Tag'
    _order = 'name'

    name = fields.Char(string="Tag Name", required=True)
    color = fields.Integer(string="Color Index")
    active = fields.Boolean(string="Active", default=True)
```

Error 24: Wrong Related Field Implementation

Symptoms: Related fields not updating, circular dependency errors

Problem:

python

```
# ❌ WRONG - Incorrect related field setup
class EstateProperty(models.Model):
    _name = 'estate.property'

    agent_id = fields.Many2one('res.partner', string="Agent")
    # Wrong: incorrect path, missing store parameter
    agent_phone = fields.Char(related='agent_id.wrong_field')
    agent_email = fields.Char(related='agent_id.email')  # No store, always computed
```

Solution:

python

```
# ✅ CORRECT - Proper related field implementation
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    agent_id = fields.Many2one(
        'res.partner',
        string="Agent",
        domain="[('is_company', '=', False)]"
    )
```

```
# Related fields with proper configuration
agent_phone = fields.Char(
    related='agent_id.phone',
    string="Agent Phone",
    store=True,     # Store for performance and searching
    readonly=True   # Usually readonly for related fields
)

agent_email = fields.Char(
    related='agent_id.email',
    string="Agent Email",
    store=True,
    readonly=True
)

# Related field through multiple levels
agent_country_id = fields.Many2one(
    related='agent_id.country_id',
    string="Agent Country",
    store=True,
    readonly=True
)

# Related field with custom string and help
company_currency_id = fields.Many2one(
    related='company_id.currency_id',
    string="Currency",
    store=True,
    readonly=True,
    help="Currency used for this property's pricing"
)
```

Error 25: Improper Selection Field Definition

Symptoms: Selection values not appearing, validation errors

Problem:

python

```
# ❌ WRONG - Incorrect selection field setup
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: Selection as string instead of list of tuples
    state = fields.Selection('draft,published,sold', string="Status")

    # Wrong: No default value, inconsistent key format
    priority = fields.Selection([
        ('Low', 'Low Priority'),      # Keys should be lowercase
        ('Medium', 'Medium'),         # Inconsistent value format
        ('HIGH', 'High Priority')     # Inconsistent key format
    ], string="Priority")
```

Solution:

python

```
# ✅ CORRECT - Proper selection field definition
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    state = fields.Selection([
        ('draft', 'Draft'),
        ('published', 'Published'),
        ('offer_received', 'Offer Received'),
        ('offer_accepted', 'Offer Accepted'),
        ('sold', 'Sold'),
        ('canceled', 'Canceled')
```

```
], string="Status", default='draft', required=True, tracking=True)

priority = fields.Selection([
    ('low', 'Low Priority'),
    ('medium', 'Medium Priority'),
    ('high', 'High Priority'),
    ('urgent', 'Urgent')
], string="Priority", default='medium', help="Property sale priority")

property_condition = fields.Selection([
    ('new', 'New Construction'),
    ('excellent', 'Excellent'),
    ('good', 'Good'),
    ('fair', 'Fair'),
    ('needs_work', 'Needs Work')
], string="Condition", required=True)

# Selection with method for dynamic values
available_financing = fields.Selection(
    selection='_get_financing_options',
    string="Available Financing"
)

@api.model
def _get_financing_options(self):
    """Dynamic selection options"""
    return [
        ('cash', 'Cash Only'),
        ('conventional', 'Conventional Loan'),
        ('fha', 'FHA Loan'),
        ('va', 'VA Loan'),
```

```
        ('owner_financing', 'Owner Financing')
    ]
```

Error 26: Binary Field Misuse

Symptoms: File upload not working, storage issues, performance problems

Problem:

python

```
# ❌ WRONG - Incorrect binary field usage
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: No filename field, no attachment handling
    brochure = fields.Binary(string="Brochure")

    # Wrong: Storing large files without attachment
    floor_plan = fields.Binary(string="Floor Plan")
```

Solution:

python

```
# ✅ CORRECT - Proper binary field implementation
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    # Binary field with filename
    brochure = fields.Binary(
        string="Property Brochure",
        help="Upload property brochure (PDF format recommended)"
    )
    brochure_filename = fields.Char(string="Brochure Filename")

    # Image field (special binary field for images)
    main_image = fields.Image(
```

```
    string="Main Property Image",
    max_width=1920,
    max_height=1080,
    help="Main property photo"
)

# Multiple images using One2many to attachment model
image_ids = fields.One2many(
    'ir.attachment',
    'res_id',
    domain=[('res_model', '=', 'estate.property'), ('mimetype', 'like', 'image/%')],
    string="Property Images"
)

# Floor plan as attachment for better storage
floor_plan_attachment_id = fields.Many2one(
    'ir.attachment',
    string="Floor Plan",
    domain="[('res_model', '=', 'estate.property'), ('res_id', '=', id)]"
)

@api.model
def create(self, vals):
    record = super().create(vals)
    # Handle file attachments after creation
    if vals.get('brochure') and vals.get('brochure_filename'):
        record._create_brochure_attachment(vals['brochure'], vals['brochure_filename'])
    return record

def _create_brochure_attachment(self, file_data, filename):
    """Create attachment for brochure"""
```

```
        self.env['ir.attachment'].create({
            'name': filename,
            'datas': file_data,
            'res_model': self._name,
            'res_id': self.id,
            'type': 'binary'
        })
```

Error 27: Date/Datetime Field Issues

Symptoms: Timezone problems, date validation errors, format issues

Problem:

python

```
# ❌ WRONG - Incorrect date/datetime usage
from datetime import datetime

class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: Using datetime.now() directly as default
    listing_date = fields.Date(default=datetime.now())

    # Wrong: No validation on date ranges
    available_from = fields.Date(string="Available From")
    available_until = fields.Date(string="Available Until")
```

Solution:

python

```
# ✅ CORRECT - Proper date/datetime field usage
from datetime import timedelta

class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'
```

```
# Proper date field with dynamic default
listing_date = fields.Date(
    string="Listing Date",
    default=fields.Date.today,  # Method reference, not call
    required=True,
    index=True
)

# Date field with computed default
available_from = fields.Date(
    string="Available From",
    default=lambda self: fields.Date.today() + timedelta(days=30),
    help="Date when property becomes available"
)

available_until = fields.Date(
    string="Available Until",
    help="Last date property is available"
)

# Datetime field for precise timestamps
last_viewed = fields.Datetime(
    string="Last Viewed",
    readonly=True,
    help="When this property was last viewed by a potential buyer"
)

# Computed datetime field
expires_on = fields.Datetime(
    string="Listing Expires",
```

```
    compute='_compute_expires_on',
    store=True
)

@api.depends('listing_date')
def _compute_expires_on(self):
    for record in self:
        if record.listing_date:
            # Convert date to datetime and add 90 days
            expire_date = record.listing_date + timedelta(days=90)
            record.expires_on = fields.Datetime.to_datetime(expire_date)
        else:
            record.expires_on = False

# Date validation constraint
@api.constrains('available_from', 'available_until')
def _check_availability_dates(self):
    for record in self:
        if record.available_from and record.available_until:
            if record.available_from > record.available_until:
                raise ValidationError(
                    "Available From date must be before Available Until date!"
                )
        if record.available_from and record.available_from < fields.Date.today():
            raise ValidationError(
                "Available From date cannot be in the past!"
            )
```

Error 28: Monetary Field Misconfiguration

Symptoms: Currency not displaying, wrong decimal places, conversion issues

Problem:

python

```
# ❌ WRONG - Incorrect monetary field setup
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: Using Float for currency without proper digits
    price = fields.Float(string="Price")

    # Wrong: Missing currency field for Monetary field
    asking_price = fields.Monetary(string="Asking Price")
```

Solution:

python

```
# ✅ CORRECT - Proper monetary field configuration
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    # Currency field (required for Monetary fields)
    currency_id = fields.Many2one(
        'res.currency',
        string="Currency",
        default=lambda self: self.env.company.currency_id,
        required=True
    )

    # Monetary field with proper currency reference
    asking_price = fields.Monetary(
        string="Asking Price",
        currency_field='currency_id',
        required=True,
        help="Initial asking price for the property"
    )
```

```
selling_price = fields.Monetary(
    string="Selling Price",
    currency_field='currency_id',
    readonly=True,
    copy=False,
    help="Final selling price"
)

# Alternative: Float with proper digits for currency
commission_amount = fields.Float(
    string="Commission Amount",
    digits='Product Price',  # Uses decimal precision
    help="Agent commission amount"
)

# Computed monetary field
total_value = fields.Monetary(
    string="Total Property Value",
    currency_field='currency_id',
    compute='_compute_total_value',
    store=True
)

@api.depends('asking_price', 'commission_amount')
def _compute_total_value(self):
    for record in self:
        record.total_value = record.asking_price + record.commission_amount

# Method to handle currency conversion
def convert_price_to_company_currency(self):
```

```
        """Convert property price to company currency"""
        company_currency = self.env.company.currency_id
        if self.currency_id != company_currency:
            converted_price = self.currency_id._convert(
                self.asking_price,
                company_currency,
                self.env.company,
                fields.Date.today()
            )
            return converted_price
        return self.asking_price
```

Error 29: Html Field Security Issues

Symptoms: XSS vulnerabilities, content not displaying, sanitization problems

Problem:

python

```
# ❌ WRONG - Html field without proper sanitization
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: No sanitization, security risk
    description = fields.Html(string="Description")

    # Wrong: All sanitization disabled
    marketing_content = fields.Html(
        string="Marketing Content",
        sanitize=False  # Dangerous!
    )
```

Solution:

python

```
# ✅ CORRECT - Secure Html field configuration
class EstateProperty(models.Model):
```

```
_name = 'estate.property'
_description = 'Real Estate Property'

# Secure Html field with sanitization
description = fields.Html(
    string="Property Description",
    sanitize=True,            # Default, but explicit is better
    sanitize_tags=True,       # Remove dangerous tags
    sanitize_attributes=True, # Remove dangerous attributes
    sanitize_style=True,      # Remove dangerous styles
    strip_style=False,        # Keep safe styles
    strip_classes=False       # Keep CSS classes
)

# Marketing content with controlled sanitization
marketing_content = fields.Html(
    string="Marketing Content",
    sanitize=True,
    help="Rich text content for property marketing"
)

# Internal notes with minimal sanitization (admin only)
internal_notes = fields.Html(
    string="Internal Notes",
    sanitize=True,
    groups="base.group_system",  # Restrict to system users
    help="Internal notes - only visible to system administrators"
)

# Plain text alternative for simple descriptions
short_description = fields.Text(
```

```
    string="Short Description",
    help="Plain text description (no HTML formatting)"
)

@api.constrains('description')
def _check_description_length(self):
    """Ensure description is not too long"""
    for record in self:
        if record.description:
            # Strip HTML tags for length check
            import re
            plain_text = re.sub('<[^<]+?>', '', record.description)
            if len(plain_text) > 5000:
                raise ValidationError(
                    "Description is too long. Maximum 5000 characters allowed."
                )
```

Error 30: Reference Field Misconfiguration

Symptoms: Reference field not working, type selection issues

Problem:

python

```
# ❌ WRONG - Incorrect Reference field setup
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Wrong: No selection models specified
    related_document = fields.Reference(string="Related Document")

    # Wrong: Invalid model references
    reference_field = fields.Reference([
        ('nonexistent.model', 'Nonexistent'),
        ('res.partner')  # Missing label
```

```
    ], string="Reference")
```

Solution:

python

```python
# ✅ CORRECT - Proper Reference field configuration
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    # Reference field with valid model selection
    related_document = fields.Reference(
        selection=[
            ('sale.order', 'Sale Order'),
            ('account.move', 'Invoice'),
            ('project.project', 'Project'),
            ('res.partner', 'Contact')
        ],
        string="Related Document",
        help="Link to related document (sale order, invoice, etc.)"
    )

    # Reference field with dynamic selection
    contact_reference = fields.Reference(
        selection='_get_contact_models',
        string="Contact Reference",
        help="Reference to contact-related records"
    )

    @api.model
    def _get_contact_models(self):
        """Dynamic selection for reference field"""
        return [
```

```
        ('res.partner', 'Contact'),
        ('res.users', 'User'),
        ('hr.employee', 'Employee')
    ]

    # Method to handle reference field operations
    def get_related_document_info(self):
        """Get information about related document"""
        if self.related_document:
            model_name = self.related_document._name
            record_id = self.related_document.id
            return {
                'model': model_name,
                'id': record_id,
                'name': self.related_document.display_name
            }
        return {}

    def set_related_document(self, model_name, record_id):
        """Set reference field programmatically"""
        if model_name and record_id:
            self.related_document = f"{model_name},{record_id}"
```

View and XML Issues (Errors 31-40) {#view-errors}

Error 31: XML Syntax and Structure Errors

Symptoms: ParseError, views not loading, malformed XML

Problem:

xml

```
<!-- ❌ WRONG - Multiple XML syntax errors -->
<odoo>
<data>
```

```
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form>
      <sheet>
        <group>
          <field name="name">   <!-- Missing closing tag -->
          <field name="price" />
        </group>
      </sheet>
    </form>
  </field>
<!-- Missing closing record tag -->
</data>
</odoo>
```

Solution:

xml

```
<!-- ✅ CORRECT - Proper XML structure -->
<odoo>
  <data>
    <record id="view_estate_property_form" model="ir.ui.view">
      <field name="name">estate.property.form</field>
      <field name="model">estate.property</field>
      <field name="priority" eval="10"/>
      <field name="arch" type="xml">
        <form string="Property Details">
          <header>
            <button name="action_publish" type="object" string="Publish"
                class="btn-primary" attrs="{'invisible': [('state', '!=', 'draft')]}"/>
            <field name="state" widget="statusbar" statusbar_visible="draft,published,sold"/>
```

```
</header>
<sheet>
  <div class="oe_title">
    <h1>
      <field name="name" placeholder="Property Name"/>
    </h1>
  </div>
  <group>
    <group string="Basic Information">
      <field name="property_type_id"/>
      <field name="asking_price" widget="monetary"/>
      <field name="currency_id" invisible="1"/>
    </group>
    <group string="Details">
      <field name="bedrooms"/>
      <field name="bathrooms"/>
      <field name="living_area"/>
    </group>
  </group>
  <notebook>
    <page string="Description">
      <field name="description" widget="html"/>
    </page>
    <page string="Offers">
      <field name="offer_ids">
        <tree editable="bottom">
          <field name="partner_id"/>
          <field name="price"/>
          <field name="status"/>
        </tree>
      </field>
```

```
                </page>
            </notebook>
        </sheet>
        <div class="oe_chatter">
            <field name="message_follower_ids"/>
            <field name="activity_ids"/>
            <field name="message_ids"/>
        </div>
    </form>
  </field>
 </record>
</data>
</odoo>
```

Error 32: Incorrect Field References in Views

Symptoms: Field not found errors, views not rendering properly

Problem:

xml

```
<!-- ❌ WRONG - Field names don't match model definition -->
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form>
      <field name="title"/>        <!-- Model has 'name' not 'title' -->
      <field name="cost"/>         <!-- Model has 'price' not 'cost' -->
      <field name="property_type"/>  <!-- Model has 'property_type_id' -->
      <field name="nonexistent"/>    <!-- Field doesn't exist at all -->
    </form>
  </field>
</record>
```

Solution:

xml

```xml
<!-- ✅ CORRECT - Field names match model definition -->
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form string="Property Details">
      <sheet>
        <group>
          <field name="name" string="Property Title"/>
          <field name="asking_price" widget="monetary"/>
          <field name="currency_id" invisible="1"/>
          <field name="property_type_id"/>
          <field name="bedrooms"/>
          <field name="bathrooms"/>
          <field name="living_area" string="Living Area (sqm)"/>
          <field name="state"/>
        </group>
      </sheet>
    </form>
  </field>
</record>

<!-- List view with proper field references -->
<record id="view_estate_property_tree" model="ir.ui.view">
  <field name="name">estate.property.tree</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <tree string="Properties" decoration-info="state == 'published'"
        decoration-success="state == 'sold'">
      <field name="name"/>
```

```
        <field name="property_type_id"/>
        <field name="asking_price" widget="monetary"/>
        <field name="currency_id" invisible="1"/>
        <field name="bedrooms"/>
        <field name="living_area"/>
        <field name="state"/>
      </tree>
    </field>
</record>
```

Error 33: Incorrect Widget Usage

Symptoms: Fields not displaying correctly, widgets not working

Problem:

xml

```xml
<!-- ❌ WRONG - Incorrect widget usage -->
<record id="view_estate_property_form" model="ir.ui.view">
   <field name="name">estate.property.form</field>
   <field name="model">estate.property</field>
   <field name="arch" type="xml">
      <form>
         <!-- Wrong: monetary widget without currency_field -->
         <field name="price" widget="monetary"/>

         <!-- Wrong: image widget on non-binary field -->
         <field name="name" widget="image"/>

         <!-- Wrong: many2many_tags on Many2one field -->
         <field name="property_type_id" widget="many2many_tags"/>

         <!-- Wrong: statusbar on non-selection field -->
         <field name="price" widget="statusbar"/>
      </form>
```

```
    </field>
</record>
```

Solution:

xml

```
<!-- ✅ CORRECT - Proper widget usage -->
<record id="view_estate_property_form" model="ir.ui.view">
    <field name="name">estate.property.form</field>
    <field name="model">estate.property</field>
    <field name="arch" type="xml">
        <form string="Property Details">
            <header>
                <!-- Statusbar widget for Selection field -->
                <field name="state" widget="statusbar"
                       statusbar_visible="draft,published,sold"/>
            </header>
            <sheet>
                <div class="oe_button_box" name="button_box">
                    <!-- Stat button for counters -->
                    <button class="oe_stat_button" type="object"
                            name="action_view_offers" icon="fa-handshake-o">
                        <field string="Offers" name="offer_count" widget="statinfo"/>
                    </button>
                </div>

                <!-- Image widget for binary field -->
                <field name="main_image" widget="image" class="oe_avatar"/>

                <group>
                    <group>
                        <!-- Monetary widget with currency field -->
                        <field name="asking_price" widget="monetary"
```

```
            options="{'currency_field': 'currency_id'}"/>
        <field name="currency_id" invisible="1"/>

        <!-- Many2one with proper domain -->
        <field name="property_type_id"
            options="{'no_create': True, 'no_open': True}"/>
    </group>
    <group>
        <!-- Integer with handle widget for ordering -->
        <field name="sequence" widget="handle"/>

        <!-- Boolean with toggle widget -->
        <field name="is_featured" widget="boolean_toggle"/>

        <!-- Date with date widget -->
        <field name="available_from" widget="date"/>
    </group>
</group>

<!-- Many2many with tags widget -->
<field name="tag_ids" widget="many2many_tags"
    options="{'color_field': 'color', 'no_create_edit': True}"/>

<!-- HTML field with html widget -->
<field name="description" widget="html"/>

<!-- One2many with tree widget -->
<field name="offer_ids">
    <tree editable="bottom">
        <field name="partner_id"/>
        <field name="price" widget="monetary"/>
```

```
                <field name="status"/>
              </tree>
            </field>
          </sheet>
        </form>
    </field>
</record>
```

Error 34: Incorrect Action Configuration

Symptoms: Actions not working, views not opening, menu items not functional

Problem:

xml

```
<!-- ❌ WRONG - Incomplete or incorrect action configuration -->
<record id="action_estate_property" model="ir.actions.act_window">
    <field name="name">Properties</field>
    <!-- Missing res_model -->
    <field name="view_mode">tree,form</field>
    <!-- Missing view references -->
</record>

<!-- Wrong menu configuration -->
<menuitem id="menu_estate_property" name="Properties"
          action="nonexistent_action"/>  <!-- Action doesn't exist -->
```

Solution:

xml

```
<!-- ✅ CORRECT - Complete action configuration -->
<record id="action_estate_property" model="ir.actions.act_window">
    <field name="name">Properties</field>
    <field name="res_model">estate.property</field>
    <field name="view_mode">tree,form,kanban</field>
    <field name="view_id" ref="view_estate_property_tree"/>
    <field name="search_view_id" ref="view_estate_property_search"/>
```

```
    <field name="context">{
        'search_default_available': 1,
        'search_default_group_by_type': 1
    }</field>
    <field name="domain">[]</field>
    <field name="help" type="html">
        <p class="o_view_nocontent_smiling_face">
            Create your first property listing!
        </p>
        <p>
            Click the create button to add a new property to your real estate portfolio.
        </p>
    </field>
    <field name="limit">80</field>
</record>

<!-- Multiple actions for different contexts -->
<record id="action_estate_property_published" model="ir.actions.act_window">
    <field name="name">Published Properties</field>
    <field name="res_model">estate.property</field>
    <field name="view_mode">kanban,tree,form</field>
    <field name="domain">[('state', '=', 'published')]</field>
    <field name="context">{'default_state': 'published'}</field>
</record>

<!-- Action with specific views -->
<record id="action_estate_property_report" model="ir.actions.act_window">
    <field name="name">Property Report</field>
    <field name="res_model">estate.property</field>
    <field name="view_mode">tree</field>
    <field name="view_id" ref="view_estate_property_tree_report"/>
```

```
    <field name="context">{'group_by': ['property_type_id', 'state']}</field>
</record>

<!-- Proper menu structure -->
<menuitem id="menu_estate_root" name="Real Estate" sequence="10"
        web_icon="real_estate,static/description/icon.png"/>

<menuitem id="menu_estate_properties" name="Properties"
        parent="menu_estate_root" sequence="10"/>

<menuitem id="menu_estate_property_all" name="All Properties"
        parent="menu_estate_properties" sequence="10"
        action="action_estate_property"/>

<menuitem id="menu_estate_property_published" name="Published"
        parent="menu_estate_properties" sequence="20"
        action="action_estate_property_published"/>

<!-- Configuration menu with groups -->
<menuitem id="menu_estate_config" name="Configuration"
        parent="menu_estate_root" sequence="100"
        groups="base.group_system"/>
```

Error 35: Search View Configuration Issues

Symptoms: Search not working, filters not appearing, grouping issues

Problem:

xml

```
<!-- ❌ WRONG - Incomplete search view -->
<record id="view_estate_property_search" model="ir.ui.view">
    <field name="name">estate.property.search</field>
    <field name="model">estate.property</field>
    <field name="arch" type="xml">
```

```
    <search>
      <!-- No field definitions -->
      <filter string="Available" domain="[('state', '=', 'available')]"/>
      <!-- Wrong domain syntax -->
      <filter string="Expensive" domain="price > 100000"/>  <!-- Missing brackets -->
    </search>
  </field>
</record>
```

Solution:

xml

```
<!-- ✅ CORRECT - Complete search view configuration -->
<record id="view_estate_property_search" model="ir.ui.view">
  <field name="name">estate.property.search</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <search string="Search Properties">
      <!-- Searchable fields -->
      <field name="name" string="Name"
          filter_domain="['|', ('name', 'ilike', self), ('description', 'ilike', self)]"/>
      <field name="property_type_id"/>
      <field name="agent_id"/>
      <field name="asking_price"/>

      <!-- Predefined filters -->
      <separator/>
      <filter string="Available" name="available"
          domain="[('state', '=', 'published')]"/>
      <filter string="Sold" name="sold"
          domain="[('state', '=', 'sold')]"/>
      <filter string="My Properties" name="my_properties"
          domain="[('agent_id.user_id', '=', uid)]"/>
```

```
        <!-- Price filters -->
        <separator/>
        <filter string="Under $500K" name="under_500k"
              domain="[('asking_price', '&lt;', 500000)]"/>
        <filter string="$500K - $1M" name="mid_range"
              domain="[('asking_price', '&gt;=', 500000), ('asking_price', '&lt;', 1000000)]"/>
        <filter string="Over $1M" name="luxury"
              domain="[('asking_price', '&gt;=', 1000000)]"/>

        <!-- Date filters -->
        <separator/>
        <filter string="Listed This Week" name="this_week"
              domain="[('create_date', '&gt;=', (context_today() -
datetime.timedelta(days=7)).strftime('%Y-%m-%d'))]"/>
        <filter string="Listed This Month" name="this_month"
              domain="[('create_date', '&gt;=', (context_today().replace(day=1)).strftime('%Y-%m-
%d'))]"/>

        <!-- Group by options -->
        <group expand="0" string="Group By">
           <filter string="Property Type" name="group_by_type"
                 context="{'group_by': 'property_type_id'}"/>
           <filter string="Agent" name="group_by_agent"
                 context="{'group_by': 'agent_id'}"/>
           <filter string="Status" name="group_by_status"
                 context="{'group_by': 'state'}"/>
           <filter string="Creation Month" name="group_by_month"
                 context="{'group_by': 'create_date:month'}"/>
        </group>

        <!-- Search panel for faceted search -->
```

```
        <searchpanel>
            <field name="property_type_id" icon="fa-building" enable_counters="1"/>
            <field name="state" select="multi" icon="fa-tag"/>
        </searchpanel>
    </search>
  </field>
</record>
```

Error 36: Kanban View Implementation Issues

Symptoms: Kanban not displaying, cards malformed, drag-drop not working

Problem:

xml

```
<!-- ❌ WRONG - Incomplete kanban view -->
<record id="view_estate_property_kanban" model="ir.ui.view">
  <field name="name">estate.property.kanban</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <kanban>
      <!-- Missing field definitions -->
      <templates>
        <t t-name="kanban-box">
          <div class="oe_kanban_card">
            <!-- Using undefined fields -->
            <field name="title"/>  <!-- Field doesn't exist -->
            <field name="cost"/>   <!-- Field doesn't exist -->
          </div>
        </t>
      </templates>
    </kanban>
  </field>
</record>
```

Solution:

xml

```xml
<!-- ✅ CORRECT - Complete kanban view -->
<record id="view_estate_property_kanban" model="ir.ui.view">
  <field name="name">estate.property.kanban</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <kanban default_group_by="state" class="o_kanban_mobile"
        quick_create="false" records_draggable="true">

      <!-- Field definitions for use in templates -->
      <field name="id"/>
      <field name="name"/>
      <field name="asking_price"/>
      <field name="currency_id"/>
      <field name="property_type_id"/>
      <field name="main_image"/>
      <field name="state"/>
      <field name="agent_id"/>
      <field name="bedrooms"/>
      <field name="bathrooms"/>
      <field name="living_area"/>

      <!-- Templates -->
      <templates>
        <t t-name="kanban-box">
          <div class="oe_kanban_card oe_kanban_global_click">
            <div class="o_kanban_image">
              <img t-att-src="kanban_image('estate.property', 'main_image', record.id.raw_value)"
                alt="Property" class="o_kanban_image_inner"/>
            </div>
```

```
<div class="oe_kanban_details">
  <div class="o_kanban_record_top">
    <div class="o_kanban_record_headings">
      <strong class="o_kanban_record_title">
        <span><t t-esc="record.name.value"/></span>
      </strong>
      <small class="o_kanban_record_subtitle text-muted">
        <t t-esc="record.property_type_id.value"/>
      </small>
    </div>

    <span class="o_kanban_record_title">
      <field name="asking_price" widget="monetary"/>
    </span>
  </div>

  <div class="o_kanban_record_body">
    <div class="row">
      <div class="col-6" t-if="record.bedrooms.raw_value">
        <i class="fa fa-bed"/> <t t-esc="record.bedrooms.value"/> bed
      </div>
      <div class="col-6" t-if="record.bathrooms.raw_value">
        <i class="fa fa-bath"/> <t t-esc="record.bathrooms.value"/> bath
      </div>
    </div>
    <div t-if="record.living_area.raw_value">
      <i class="fa fa-home"/> <t t-esc="record.living_area.value"/> sqm
    </div>
  </div>

  <div class="o_kanban_record_bottom">
```

```
                    <div class="oe_kanban_bottom_left">
                      <span class="badge badge-pill"
                          t-att-class="record.state.raw_value === 'published' ? 'badge-success' :
'badge-secondary'">
                        <t t-esc="record.state.value"/>
                      </span>
                    </div>
                    <div class="oe_kanban_bottom_right">
                      <img t-att-src="kanban_image('res.partner', 'image_128',
record.agent_id.raw_value)"
                          t-att-title="record.agent_id.value"
                          class="oe_kanban_avatar o_image_24_cover rounded-circle"/>
                    </div>
                  </div>
                </div>
              </div>
            </t>
          </templates>
        </kanban>
      </field>
</record>
```

Error 37: Form View Layout Problems

Symptoms: Poor form layout, fields not aligned, responsive issues

Problem:

xml

```xml
<!-- ❌ WRONG - Poor form layout -->
<record id="view_estate_property_form" model="ir.ui.view">
  <field name="name">estate.property.form</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <form>
      <!-- No sheet wrapper, poor organization -->
```

```
        <field name="name"/>
        <field name="price"/>
        <field name="type"/>
        <!-- No grouping, all fields in single column -->
        <field name="bedrooms"/>
        <field name="bathrooms"/>
        <field name="description"/>
      </form>
    </field>
</record>
```

Solution:

xml

```
<!-- ✅ CORRECT - Well-structured form layout -->
<record id="view_estate_property_form" model="ir.ui.view">
    <field name="name">estate.property.form</field>
    <field name="model">estate.property</field>
    <field name="arch" type="xml">
      <form string="Property Details">
        <!-- Header for status and action buttons -->
        <header>
          <button name="action_publish" type="object" string="Publish"
                  class="btn-primary"
                  attrs="{'invisible': [('state', '!=', 'draft')]}"/>
          <button name="action_mark_sold" type="object" string="Mark as Sold"
                  class="btn-success"
                  attrs="{'invisible': [('state', '!=', 'published')]}"/>
          <field name="state" widget="statusbar"
                 statusbar_visible="draft,published,sold"/>
        </header>

        <sheet>
```

```
<!-- Button box for stat buttons -->
<div class="oe_button_box" name="button_box">
  <button class="oe_stat_button" type="object" name="action_view_offers"
        icon="fa-handshake-o">
    <field string="Offers" name="offer_count" widget="statinfo"/>
  </button>
  <button class="oe_stat_button" type="object" name="action_view_visits"
        icon="fa-eye">
    <field string="Visits" name="visit_count" widget="statinfo"/>
  </button>
</div>

<!-- Main image -->
<field name="main_image" widget="image" class="oe_avatar"
    options="{'preview_image': 'main_image', 'size': [150, 150]}"/>

<!-- Title section -->
<div class="oe_title">
  <label for="name" class="oe_edit_only"/>
  <h1>
    <field name="name" placeholder="e.g. Beautiful Family Home"/>
  </h1>
  <h3>
    <field name="property_type_id" placeholder="Property Type"/>
  </h3>
</div>

<!-- Main form groups -->
<group>
  <group string="Basic Information">
    <field name="asking_price" widget="monetary"/>
```

```
        <field name="currency_id" invisible="1"/>
        <field name="agent_id"/>
        <field name="listing_date"/>
        <field name="available_from"/>
    </group>

    <group string="Property Details">
        <field name="bedrooms"/>
        <field name="bathrooms"/>
        <field name="living_area" string="Living Area (sqm)"/>
        <field name="lot_size" string="Lot Size (sqm)"/>
        <field name="year_built"/>
    </group>
</group>

<!-- Tags field -->
<field name="tag_ids" widget="many2many_tags"
       options="{'color_field': 'color', 'no_create_edit': True}"/>

<!-- Notebook for organized sections -->
<notebook>
    <page string="Description" name="description">
        <group>
            <field name="description" widget="html" nolabel="1"
                   placeholder="Enter detailed property description..."/>
        </group>
    </page>

    <page string="Features" name="features">
        <group>
            <group string="Interior Features">
```

```
            <field name="has_garage"/>
            <field name="has_pool"/>
            <field name="has_fireplace"/>
            <field name="has_air_conditioning"/>
        </group>
        <group string="Exterior Features">
            <field name="has_garden"/>
            <field name="has_balcony"/>
            <field name="has_terrace"/>
            <field name="parking_spaces"/>
        </group>
    </group>
</page>

<page string="Offers" name="offers">
    <field name="offer_ids" mode="tree">
        <tree decoration-success="status == 'accepted'"
              decoration-danger="status == 'rejected'"
              editable="bottom">
            <field name="partner_id"/>
            <field name="price" widget="monetary"/>
            <field name="offer_date"/>
            <field name="status"/>
            <button name="action_accept" type="object"
                    icon="fa-check" string="Accept"
                    attrs="{'invisible': [('status', '!=', 'pending')]}"/>
            <button name="action_reject" type="object"
                    icon="fa-times" string="Reject"
                    attrs="{'invisible': [('status', '!=', 'pending')]}"/>
        </tree>
    </field>
```

```
            </page>

            <page string="Documents" name="documents">
              <field name="document_ids">
                <tree>
                  <field name="name"/>
                  <field name="document_type"/>
                  <field name="upload_date"/>
                </tree>
              </field>
            </page>
          </notebook>
        </sheet>

        <!-- Chatter for communication -->
        <div class="oe_chatter">
          <field name="message_follower_ids"/>
          <field name="activity_ids"/>
          <field name="message_ids"/>
        </div>
      </form>
    </field>
</record>
```

Error 38: Tree View Configuration Issues

Symptoms: Columns not displaying correctly, sorting issues, decoration problems

Problem:

xml

```
<!-- ❌ WRONG - Poor tree view configuration -->
<record id="view_estate_property_tree" model="ir.ui.view">
  <field name="name">estate.property.tree</field>
  <field name="model">estate.property</field>
```

```
  <field name="arch" type="xml">
    <tree>
      <!-- No field labels, poor column selection -->
      <field name="name"/>
      <field name="price"/>
      <field name="create_date"/>
      <!-- No decoration, all records look the same -->
    </tree>
  </field>
</record>
```

Solution:

xml

```
<!-- ✅ CORRECT - Well-configured tree view -->
<record id="view_estate_property_tree" model="ir.ui.view">
  <field name="name">estate.property.tree</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <tree string="Properties"
        decoration-info="state == 'draft'"
        decoration-success="state == 'published'"
        decoration-warning="state == 'offer_received'"
        decoration-muted="state == 'sold'"
        multi_edit="1"
        sample="1">

      <!-- Handle for manual sorting if sequence field exists -->
      <field name="sequence" widget="handle"/>

      <!-- Main identification fields -->
      <field name="name" string="Property Name"/>
      <field name="property_type_id"/>
```

```
<!-- Price information -->
<field name="asking_price" widget="monetary" sum="Total Value"/>
<field name="currency_id" invisible="1"/>

<!-- Property details -->
<field name="bedrooms" string="Beds"/>
<field name="bathrooms" string="Baths"/>
<field name="living_area" string="Living Area (sqm)" sum="Total Area"/>

<!-- Business information -->
<field name="agent_id"/>
<field name="listing_date"/>

<!-- Status with color coding -->
<field name="state" widget="badge"
    decoration-info="state == 'draft'"
    decoration-success="state == 'published'"
    decoration-warning="state == 'offer_received'"/>

<!-- Optional fields that can be hidden/shown -->
<field name="available_from" optional="hide"/>
<field name="year_built" optional="hide"/>
<field name="lot_size" optional="hide"/>

<!-- Computed fields -->
<field name="offer_count" string="Offers" optional="show"/>
<field name="days_on_market" optional="hide"/>

<!-- Action buttons -->
<button name="action_publish" type="object"
```

```
            icon="fa-globe" string="Publish"
            attrs="{'invisible': [('state', '!=', 'draft')]}"/>
        <button name="action_mark_sold" type="object"
            icon="fa-check" string="Mark Sold"
            attrs="{'invisible': [('state', '!=', 'published')]}"/>
    </tree>
  </field>
</record>

<!-- Alternative tree view for different contexts -->
<record id="view_estate_property_tree_simple" model="ir.ui.view">
  <field name="name">estate.property.tree.simple</field>
  <field name="model">estate.property</field>
  <field name="arch" type="xml">
    <tree string="Properties" create="false" delete="false" edit="false">
      <field name="name"/>
      <field name="asking_price" widget="monetary"/>
      <field name="currency_id" invisible="1"/>
      <field name="state"/>
    </tree>
  </field>
</record>
```

Error 39: Menu Structure Problems

Symptoms: Menus not appearing, wrong hierarchy, access issues

Problem:

xml

```
<!-- ❌ WRONG - Poor menu structure -->
<!-- All menus at root level, no hierarchy -->
<menuitem id="menu_estate" name="Estate" action="action_estate_property"/>
<menuitem id="menu_types" name="Types" action="action_estate_type"/>
<menuitem id="menu_agents" name="Agents" action="action_estate_agent"/>
```

```
<!-- Missing sequence, random order -->
<menuitem id="menu_config" name="Configuration"/>
```

Solution:

xml

```
<!-- ✅ CORRECT - Proper menu hierarchy -->
<!-- Root application menu -->
<menuitem id="menu_estate_root"
        name="Real Estate"
        sequence="10"
        web_icon="real_estate,static/description/icon.png"/>

<!-- Main sections -->
<menuitem id="menu_estate_operations"
        name="Operations"
        parent="menu_estate_root"
        sequence="10"/>

<menuitem id="menu_estate_reports"
        name="Reports"
        parent="menu_estate_root"
        sequence="20"/>

<menuitem id="menu_estate_configuration"
        name="Configuration"
        parent="menu_estate_root"
        sequence="100"
        groups="base.group_system"/>

<!-- Operations submenus -->
<menuitem id="menu_estate_properties"
```

```
        name="Properties"
        parent="menu_estate_operations"
        sequence="10"
        action="action_estate_property"/>

<menuitem id="menu_estate_offers"
        name="Offers"
        parent="menu_estate_operations"
        sequence="20"
        action="action_estate_offer"/>

<menuitem id="menu_estate_visits"
        name="Property Visits"
        parent="menu_estate_operations"
        sequence="30"
        action="action_estate_visit"/>

<!-- Reports submenus -->
<menuitem id="menu_estate_property_report"
        name="Property Analysis"
        parent="menu_estate_reports"
        sequence="10"
        action="action_estate_property_report"/>

<menuitem id="menu_estate_sales_report"
        name="Sales Report"
        parent="menu_estate_reports"
        sequence="20"
        action="action_estate_sales_report"/>

<!-- Configuration submenus with access control -->
```

```
<menuitem id="menu_estate_property_types"
        name="Property Types"
        parent="menu_estate_configuration"
        sequence="10"
        action="action_estate_property_type"
        groups="real_estate.group_real_estate_manager"/>

<menuitem id="menu_estate_property_tags"
        name="Property Tags"
        parent="menu_estate_configuration"
        sequence="20"
        action="action_estate_property_tag"
        groups="real_estate.group_real_estate_manager"/>

<menuitem id="menu_estate_settings"
        name="Settings"
        parent="menu_estate_configuration"
        sequence="100"
        action="action_estate_settings"
        groups="base.group_system"/>

<!-- Alternative organization by user type -->
<menuitem id="menu_estate_agent"
        name="Agent Dashboard"
        parent="menu_estate_root"
        sequence="5"
        action="action_estate_agent_dashboard"
        groups="real_estate.group_real_estate_agent"/>

<menuitem id="menu_estate_manager"
        name="Manager Dashboard"
```

```
        parent="menu_estate_root"
        sequence="6"
        action="action_estate_manager_dashboard"
        groups="real_estate.group_real_estate_manager"/>
```

Error 40: View Inheritance Problems

Symptoms: Inherited views not working, XPath errors, position issues

Problem:

xml

```
<!-- ❌ WRONG - Incorrect view inheritance -->
<record id="view_partner_form_inherit" model="ir.ui.view">
   <field name="name">res.partner.form.inherit</field>
   <field name="model">res.partner</field>
   <!-- Missing inherit_id -->
   <field name="arch" type="xml">
      <!-- Wrong XPath syntax -->
      <xpath expr="field[@name='email']" position="after">
         <field name="is_real_estate_agent"/>
      </xpath>
   </field>
</record>
```

Solution:

xml

```
<!-- ✅ CORRECT - Proper view inheritance -->
<record id="view_partner_form_inherit_real_estate" model="ir.ui.view">
   <field name="name">res.partner.form.inherit.real.estate</field>
   <field name="model">res.partner</field>
   <field name="inherit_id" ref="base.view_partner_form"/>
   <field name="priority" eval="20"/>
   <field name="arch" type="xml">

      <!-- Add real estate tab using xpath -->
```

```
<xpath expr="//notebook" position="inside">
  <page string="Real Estate" name="real_estate"
      attrs="{'invisible': [('is_real_estate_agent', '=', False)]}">
    <group>
      <group string="Agent Information">
        <field name="is_real_estate_agent"/>
        <field name="agent_license_number"
            attrs="{'required': [('is_real_estate_agent', '=', True)]}"/>
        <field name="agent_experience_years"/>
        <field name="specialization_ids" widget="many2many_tags"/>
      </group>
      <group string="Performance">
        <field name="properties_sold_count"/>
        <field name="total_sales_amount" widget="monetary"/>
        <field name="average_deal_time"/>
      </group>
    </group>

    <!-- Properties managed by this agent -->
    <group string="Properties">
      <field name="property_ids" nolabel="1">
        <tree>
          <field name="name"/>
          <field name="asking_price" widget="monetary"/>
          <field name="state"/>
          <field name="listing_date"/>
        </tree>
      </field>
    </group>
  </page>
</xpath>
```

```
    <!-- Add field to existing group using field location -->
    <field name="category_id" position="after">
      <field name="is_real_estate_agent"/>
    </field>

    <!-- Modify existing field attributes -->
    <field name="phone" position="attributes">
      <attribute name="required">1</attribute>
      <attribute name="help">Phone number is required for real estate agents</attribute>
    </field>

    <!-- Add to button box -->
    <xpath expr="//div[@name='button_box']" position="inside">
      <button class="oe_stat_button" type="object"
              name="action_view_agent_properties" icon="fa-home"
              attrs="{'invisible': [('is_real_estate_agent', '=', False)]}">
          <field string="Properties" name="property_count" widget="statinfo"/>
      </button>
    </xpath>

    <!-- Replace entire section -->
    <xpath expr="//group[@name='sale']" position="replace">
      <group name="sale" string="Sales Information">
          <field name="customer_rank" widget="boolean_toggle"/>
          <field name="supplier_rank" widget="boolean_toggle"/>
          <field name="is_real_estate_agent" widget="boolean_toggle"/>
      </group>
    </xpath>

  </field>
```

```
</record>

<!-- Inherit tree view -->
<record id="view_partner_tree_inherit_real_estate" model="ir.ui.view">
  <field name="name">res.partner.tree.inherit.real.estate</field>
  <field name="model">res.partner</field>
  <field name="inherit_id" ref="base.view_partner_tree"/>
  <field name="arch" type="xml">
    <field name="email" position="after">
      <field name="is_real_estate_agent" string="Agent"/>
      <field name="property_count" string="Properties"
          attrs="{'invisible': [('is_real_estate_agent', '=', False)]}"/>
    </field>
  </field>
</record>

<!-- Inherit search view -->
<record id="view_partner_search_inherit_real_estate" model="ir.ui.view">
  <field name="name">res.partner.search.inherit.real.estate</field>
  <field name="model">res.partner</field>
  <field name="inherit_id" ref="base.view_res_partner_filter"/>
  <field name="arch" type="xml">

    <!-- Add search field -->
    <field name="name" position="after">
      <field name="agent_license_number"/>
    </field>

    <!-- Add filter -->
    <filter name="customer" position="after">
      <filter string="Real Estate Agents" name="real_estate_agents"
```

```
            domain="[('is_real_estate_agent', '=', True)]"/>
    </filter>

    <!-- Add group by -->
    <xpath expr="//group[@expand='0']" position="inside">
      <filter string="Agent Status" name="group_agent_status"
            context="{'group_by': 'is_real_estate_agent'}"/>
    </xpath>

  </field>
</record>
```

Security and Access Rights (Errors 41-45) {#security-errors}

Error 41: Missing ir.model.access.csv File

Symptoms: AccessError when trying to access model records

Problem:

```
# No security/ir.model.access.csv file exists
# Or file exists but is empty or incorrectly formatted
```

Solution:

csv

```
# security/ir.model.access.csv
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_estate_property_user,estate.property.user,model_estate_property,base.group_user,1,1,1,1
access_estate_property_manager,estate.property.manager,model_estate_property,real_estate.group_real_estate_manager,1,1,1,1
access_estate_property_agent,estate.property.agent,model_estate_property,real_estate.group_real_estate_agent,1,1,1,0
access_estate_property_type_user,estate.property.type.user,model_estate_property_type,base.group_user,1,0,0,0
access_estate_property_type_manager,estate.property.type.manager,model_estate_property_type,real_estate.group_real_estate_manager,1,1,1,1
access_estate_property_tag_user,estate.property.tag.user,model_estate_property_tag,base.group_user,1,0,0,0
```

access_estate_property_tag_manager,estate.property.tag.manager,model_estate_property_tag,real_estate.group_real_estate_manager,1,1,1,1

access_estate_property_offer_user,estate.property.offer.user,model_estate_property_offer,base.group_user,1,1,1,1

access_estate_property_offer_portal,estate.property.offer.portal,model_estate_property_offer,base.group_portal,1,0,1,0

Error 42: Incorrect Access Rights CSV Format

Symptoms: CSV parsing errors, access rights not loading

Problem:

csv

❌ WRONG - Incorrect CSV format

Missing header

estate_property_user,estate.property.user,model_estate_property,base.group_user,1,1,1,1

Wrong header

id;name;model_id;group_id;read;write;create;delete

access_estate_property_user;estate.property.user;model_estate_property;base.group_user;1;1;1;1

Missing columns

id,name,model_id:id,group_id:id,perm_read

access_estate_property_user,estate.property.user,model_estate_property,base.group_user,1

Solution:

csv

✅ CORRECT - Proper CSV format with exact header

id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink

access_estate_property_user,estate.property.user,model_estate_property,base.group_user,1,1,1,1

access_estate_property_public,estate.property.public,model_estate_property,,1,0,0,0

access_estate_property_manager,estate.property.manager,model_estate_property,real_estate.group_real_estate_manager,1,1,1,1

access_estate_property_type_user,estate.property.type.user,model_estate_property_type,base.group_user,1,0,0,0

access_estate_property_type_manager,estate.property.type.manager,model_estate_property_type,real_estate.group_real_estate_manager,1,1,1,1

Error 43: Security Groups Not Defined

Symptoms: Groups referenced in CSV don't exist, access rights not working

Problem:

csv

```csv
# ❌ WRONG - Referencing non-existent groups
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_estate_property_agent,estate.property.agent,model_estate_property,real_estate.group_agent,1,1,1,0
# Group 'real_estate.group_agent' doesn't exist
```

Solution:

xml

```xml
<!-- ✅ CORRECT - Define security groups first -->
<!-- security/real_estate_security.xml -->
<odoo>
  <data noupdate="1">

    <!-- Define category -->
    <record id="module_category_real_estate" model="ir.module.category">
      <field name="name">Real Estate</field>
      <field name="description">Manage real estate operations</field>
      <field name="sequence">10</field>
    </record>

    <!-- Define groups -->
    <record id="group_real_estate_agent" model="res.groups">
      <field name="name">Agent</field>
      <field name="category_id" ref="module_category_real_estate"/>
      <field name="implied_ids" eval="[(4, ref('base.group_user'))]"/>
      <field name="comment">Real estate agent with basic property management rights</field>
    </record>
```

```
    <record id="group_real_estate_manager" model="res.groups">
      <field name="name">Manager</field>
      <field name="category_id" ref="module_category_real_estate"/>
      <field name="implied_ids" eval="[(4, ref('group_real_estate_agent'))]"/>
      <field name="comment">Real estate manager with full management rights</field>
    </record>

    <record id="group_real_estate_admin" model="res.groups">
      <field name="name">Administrator</field>
      <field name="category_id" ref="module_category_real_estate"/>
      <field name="implied_ids" eval="[(4, ref('group_real_estate_manager'))]"/>
      <field name="comment">Full administrative access to real estate module</field>
    </record>

  </data>
</odoo>
```

python

```
# Update __manifest__.py to include security files in correct order
{
  'data': [
    'security/real_estate_security.xml',  # Groups first
    'security/ir.model.access.csv',       # Then access rights
    'views/estate_property_views.xml',
    # ... other files
  ],
}
```

Error 44: Record Rules (ir.rule) Conflicts

Symptoms: Users can't access records they should, or can access records they shouldn't

Problem:

xml

```
<!-- ❌ WRONG - Conflicting or incorrect record rules -->
```

```
<record id="estate_property_rule_agent" model="ir.rule">
   <field name="name">Agent can only see own properties</field>
   <field name="model_id" ref="model_estate_property"/>
   <field name="domain_force">[('agent_id.user_id', '=', user.id)]</field>
   <field name="groups" eval="[(4, ref('group_real_estate_agent'))]"/>
   <!-- Missing permission fields - applies to all operations -->
</record>

<!-- Global rule that's too restrictive -->
<record id="estate_property_rule_global" model="ir.rule">
   <field name="name">Properties must be published</field>
   <field name="model_id" ref="model_estate_property"/>
   <field name="domain_force">[('state', '=', 'published')]</field>
   <!-- No groups - applies to everyone including managers -->
</record>
```

Solution:

xml

```
<!-- ✅ CORRECT - Proper record rules configuration -->
<odoo>
   <data noupdate="1">

      <!-- Rule for agents - can see own properties and published ones -->
      <record id="estate_property_rule_agent" model="ir.rule">
         <field name="name">Agent Property Access</field>
         <field name="model_id" ref="model_estate_property"/>
         <field name="domain_force">['|',
            ('agent_id.user_id', '=', user.id),
            ('state', '=', 'published')]</field>
         <field name="groups" eval="[(4, ref('group_real_estate_agent'))]"/>
         <field name="perm_read" eval="True"/>
         <field name="perm_write" eval="True"/>
```

```
    <field name="perm_create" eval="True"/>
    <field name="perm_unlink" eval="False"/>  <!-- Agents can't delete -->
</record>

<!-- Rule for managers - can see all properties -->
<record id="estate_property_rule_manager" model="ir.rule">
    <field name="name">Manager Property Access</field>
    <field name="model_id" ref="model_estate_property"/>
    <field name="domain_force">[(1, '=', 1)]</field>  <!-- Always true -->
    <field name="groups" eval="[(4, ref('group_real_estate_manager'))]"/>
    <field name="perm_read" eval="True"/>
    <field name="perm_write" eval="True"/>
    <field name="perm_create" eval="True"/>
    <field name="perm_unlink" eval="True"/>
</record>

<!-- Multi-company rule -->
<record id="estate_property_rule_multicompany" model="ir.rule">
    <field name="name">Property Multi-Company Rule</field>
    <field name="model_id" ref="model_estate_property"/>
    <field name="global" eval="True"/>  <!-- Global rule -->
    <field name="domain_force">['|',
        ('company_id', '=', False),
        ('company_id', 'in', company_ids)]</field>
</record>

<!-- Portal user rule -->
<record id="estate_property_rule_portal" model="ir.rule">
    <field name="name">Portal Property Access</field>
    <field name="model_id" ref="model_estate_property"/>
    <field name="domain_force">[('state', '=', 'published')]</field>
```

```
        <field name="groups" eval="[(4, ref('base.group_portal'))]"/>
        <field name="perm_read" eval="True"/>
        <field name="perm_write" eval="False"/>
        <field name="perm_create" eval="False"/>
        <field name="perm_unlink" eval="False"/>
    </record>

    <!-- Offer rules -->
    <record id="estate_offer_rule_agent" model="ir.rule">
        <field name="name">Agent Offer Access</field>
        <field name="model_id" ref="model_estate_property_offer"/>
        <field name="domain_force">['|',
            ('property_id.agent_id.user_id', '=', user.id),
            ('partner_id.user_id', '=', user.id)]</field>
        <field name="groups" eval="[(4, ref('group_real_estate_agent'))]"/>
    </record>

  </data>
</odoo>
```

Error 45: Field-Level Security Issues

Symptoms: Sensitive fields visible to wrong users, missing groups on fields

Problem:

python

```
# ❌ WRONG - No field-level security on sensitive data
class EstateProperty(models.Model):
    _name = 'estate.property'

    # Sensitive financial data visible to everyone
    commission_rate = fields.Float(string="Commission Rate")
    agent_commission = fields.Monetary(string="Agent Commission")
    profit_margin = fields.Float(string="Profit Margin")
```

```
    # Internal notes visible to all users
    internal_notes = fields.Text(string="Internal Notes")
```

Solution:

python

```
# ✅ CORRECT - Proper field-level security
class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    # Financial data restricted to managers
    commission_rate = fields.Float(
        string="Commission Rate",
        groups="real_estate.group_real_estate_manager"
    )
    agent_commission = fields.Monetary(
        string="Agent Commission",
        currency_field='currency_id',
        groups="real_estate.group_real_estate_manager"
    )
    profit_margin = fields.Float(
        string="Profit Margin",
        groups="real_estate.group_real_estate_manager"
    )

    # Internal data restricted to internal users
    internal_notes = fields.Text(
        string="Internal Notes",
        groups="base.group_user"  # Only internal users
    )
```

```
    # System data restricted to system administrators
    debug_info = fields.Text(
        string="Debug Information",
        groups="base.group_system"
    )

    # Public information (no groups restriction)
    name = fields.Char(string="Property Title", required=True)
    description = fields.Html(string="Description")
    asking_price = fields.Monetary(string="Asking Price", currency_field='currency_id')

    # Agent-specific fields
    agent_notes = fields.Text(
        string="Agent Notes",
        groups="real_estate.group_real_estate_agent",
        help="Private notes for the assigned agent"
    )
xml
<!-- Field-level security in views -->
<record id="view_estate_property_form" model="ir.ui.view">
    <field name="name">estate.property.form</field>
    <field name="model">estate.property</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <group string="Basic Information">
                        <field name="name"/>
                        <field name="asking_price" widget="monetary"/>
                        <field name="currency_id" invisible="1"/>
                    </group>
```

```
            <!-- Financial group only for managers -->
            <group string="Financial Details"
                   groups="real_estate.group_real_estate_manager">
              <field name="commission_rate"/>
              <field name="agent_commission" widget="monetary"/>
              <field name="profit_margin"/>
            </group>
          </group>

          <notebook>
            <page string="Description">
              <field name="description" widget="html"/>
            </page>

            <!-- Internal notes tab restricted -->
            <page string="Internal Notes"
                  groups="base.group_user">
              <field name="internal_notes"/>
            </page>

            <!-- Debug tab for system users only -->
            <page string="Debug"
                  groups="base.group_system">
              <field name="debug_info"/>
            </page>
          </notebook>
        </sheet>
      </form>
    </field>
  </record>
```

Server and Database Issues (Errors 46-50) {#server-errors}

Error 46: Database Connection Problems

Symptoms: psycopg2 errors, connection refused, authentication failed

Problem:

bash

```
# ❌ Common database issues:
# - PostgreSQL not running
# - Wrong connection parameters
# - User doesn't exist or no permissions
# - Database doesn't exist

# Error examples:
# psycopg2.OperationalError: could not connect to server
# psycopg2.OperationalError: FATAL: database "odoo" does not exist
# psycopg2.OperationalError: FATAL: role "odoo" does not exist
```

Solution:

bash

```
# ✅ CORRECT - Database setup and troubleshooting

# 1. Check if PostgreSQL is running
sudo systemctl status postgresql
sudo systemctl start postgresql  # Start if not running

# 2. Create PostgreSQL user for Odoo
sudo -u postgres createuser -s odoo  # Create superuser
# or with specific permissions:
sudo -u postgres createuser --createdb --username postgres --no-createrole --no-superuser odoo

# 3. Set password for user (optional but recommended)
sudo -u postgres psql
```

```
ALTER USER odoo PASSWORD 'your_password';
\q

# 4. Create database
sudo -u postgres createdb -O odoo odoo_db

# 5. Test connection
psql -h localhost -U odoo -d odoo_db -W

# 6. Configure Odoo connection parameters
# In odoo.conf:
[options]
db_host = localhost
db_port = 5432
db_user = odoo
db_password = your_password
db_name = odoo_db

# 7. For development - allow local connections
# Edit /etc/postgresql/*/main/pg_hba.conf:
local   all         odoo                        trust
host    all         odoo        127.0.0.1/32        trust
host    all         odoo        ::1/128             trust

# 8. Restart PostgreSQL after config changes
sudo systemctl restart postgresql
```

Error 47: Module Update and Installation Issues

Symptoms: Module not updating, changes not reflected, installation fails

Problem:

bash

❌ *WRONG - Common module issues:*

 # - Not updating module after code changes

 # - Module not in addons path

 # - Syntax errors preventing load

 # - Cache issues

 # Starting Odoo without update after changes:

./odoo-bin --addons-path=addons -d mydb

Solution:

bash

 # ✅ CORRECT - Proper module management

 # 1. Update module after code changes

./odoo-bin --addons-path=addons -d mydb -u my_module

 # 2. Install new module

./odoo-bin --addons-path=addons -d mydb -i my_module

 # 3. Update multiple modules

./odoo-bin --addons-path=addons -d mydb -u my_module,another_module

 # 4. Development mode with auto-reload

./odoo-bin --addons-path=addons -d mydb --dev xml,reload

 # 5. Force update all modules (use with caution)

./odoo-bin --addons-path=addons -d mydb -u all

 # 6. Check addons path configuration

./odoo-bin --addons-path=addons,custom_addons,extra_addons -d mydb

 # 7. Clear Python cache if having import issues

find . -name "*.pyc" -delete

```
find . -name "__pycache__" -type d -exec rm -rf {} +

# 8. Restart with clean slate (for persistent issues)
./odoo-bin --addons-path=addons -d mydb --stop-after-init -u my_module
./odoo-bin --addons-path=addons -d mydb

# 9. Using configuration file
./odoo-bin -c /path/to/odoo.conf -u my_module

# 10. Debug mode for troubleshooting
./odoo-bin --addons-path=addons -d mydb --log-level=debug -u my_module
```

Error 48: Import and Dependency Errors

Symptoms: ImportError, ModuleNotFoundError, circular import issues

Problem:

python

```
# ❌ WRONG - Various import issues

# Incorrect Odoo imports for version 18
from openerp import models, fields, api  # Wrong for Odoo 18

# Missing dependencies in manifest
{
    'depends': ['base'],  # Missing required dependencies
}

# In code using sale module features:
class EstateProperty(models.Model):
    sale_order_id = fields.Many2one('sale.order')  # Error: sale not in depends

# Circular imports
# models/estate_property.py
```

```
from .estate_offer import EstateOffer  # Wrong import style

# models/estate_offer.py
from .estate_property import EstateProperty  # Circular dependency
```

Solution:

python

```
# ✅ CORRECT - Proper imports and dependencies

# Correct imports for Odoo 18
from odoo import models, fields, api
from odoo.exceptions import ValidationError, UserError
from odoo.tools import float_compare, float_is_zero

# Correct manifest dependencies
{
    'name': 'Real Estate',
    'depends': [
        'base',
        'mail',         # For mail.thread
        'sale',         # For sale.order integration
        'website',      # For website features
    ],
}

# Correct model definition with proper dependencies
class EstateProperty(models.Model):
    _name = 'estate.property'
    _inherit = ['mail.thread', 'mail.activity.mixin']
    _description = 'Real Estate Property'

    # Safe to reference sale.order since 'sale' is in depends
```

```
    sale_order_id = fields.Many2one('sale.order', string="Related Sale Order")

# Avoid circular imports - use string references
class EstateProperty(models.Model):
    _name = 'estate.property'

    offer_ids = fields.One2many(
        'estate.property.offer',  # String reference, no import needed
        'property_id',
        string="Offers"
    )

    def create_offer(self):
        # Access related model through env
        offer_model = self.env['estate.property.offer']
        return offer_model.create({
            'property_id': self.id,
            'price': 100000,
        })

# Proper __init__.py structure
# models/__init__.py
from . import estate_property
from . import estate_property_type
from . import estate_property_offer
# No circular imports, just module registration

# Import external libraries safely
try:
    import requests
except ImportError:
```

```
    requests = None

class EstateProperty(models.Model):
    _name = 'estate.property'

    def sync_with_external_api(self):
        if not requests:
            raise UserError("The 'requests' library is required for this feature")
        # Use requests library...
```

Error 49: Configuration and Server Issues

Symptoms: Server won't start, configuration errors, port conflicts

Problem:

bash

```
# ❌ WRONG - Common configuration issues:

# Missing or incorrect configuration
./odoo-bin  # No database specified, no addons path

# Port conflicts
./odoo-bin -p 8069  # Port already in use

# Wrong file paths
./odoo-bin --addons-path=/nonexistent/path

# Insufficient permissions
./odoo-bin --logfile=/var/log/odoo/odoo.log  # Permission denied
```

Solution:

bash

```
# ✅ CORRECT - Proper server configuration

# 1. Create proper configuration file
```

```
# /etc/odoo/odoo.conf or ~/.odoorc
[options]
addons_path = /opt/odoo/addons,/opt/odoo/custom_addons
admin_passwd = admin_password
db_host = localhost
db_port = 5432
db_user = odoo
db_password = odoo_password
dbfilter = ^mycompany_.*$
http_port = 8069
longpolling_port = 8072
logfile = /var/log/odoo/odoo.log
log_level = info
workers = 4
max_cron_threads = 2
db_maxconn = 64

# 2. Start with configuration file
./odoo-bin -c /etc/odoo/odoo.conf

# 3. Handle port conflicts
./odoo-bin -p 8070  # Use different port
# or find and kill process using port 8069:
sudo lsof -t -i:8069 | xargs sudo kill -9

# 4. Set up proper file permissions
sudo mkdir -p /var/log/odoo
sudo chown odoo:odoo /var/log/odoo
sudo chmod 755 /var/log/odoo

# 5. Development configuration
```

```
./odoo-bin \
    --addons-path=addons,custom_addons \
    --db-filter=^dev_.*$ \
    --dev=xml,reload \
    --log-level=debug \
    --workers=0  # Single worker for development

# 6. Production configuration example
./odoo-bin \
    -c /etc/odoo/odoo.conf \
    --workers=4 \
    --max-cron-threads=2 \
    --limit-memory-hard=2684354560 \
    --limit-memory-soft=2147483648 \
    --limit-time-cpu=600 \
    --limit-time-real=1200

# 7. Debug startup issues
./odoo-bin \
    --addons-path=addons \
    -d mydb \
    --log-level=debug \
    --stop-after-init  # Stop after initialization for debugging

# 8. Check configuration
./odoo-bin --help  # See all available options
./odoo-bin -c /etc/odoo/odoo.conf --test-file  # Test configuration
```

Error 50: Performance and Memory Issues

Symptoms: Slow performance, memory leaks, timeouts, high CPU usage

Problem:

python

```
# ❌ WRONG - Performance-killing patterns

# N+1 queries
class EstateProperty(models.Model):
    _name = 'estate.property'

    def get_all_agent_names(self):
        names = []
        for prop in self.search([]):
            names.append(prop.agent_id.name)  # N+1 query problem
        return names

# Loading unnecessary data
def get_property_summary(self):
    properties = self.env['estate.property'].search([])  # Loads all fields
    return len(properties)

# No indexes on frequently searched fields
class EstateProperty(models.Model):
    _name = 'estate.property'

    property_code = fields.Char(string="Code")  # No index, searched frequently

# Inefficient computed fields
@api.depends('offer_ids.price')
def _compute_best_offer(self):
    for record in self:
        if record.offer_ids:
            # Inefficient: loads all offer records
            record.best_offer = max(record.offer_ids.mapped('price'))
```

Solution:

python

```python
# ✅ CORRECT - Performance optimizations

class EstateProperty(models.Model):
    _name = 'estate.property'
    _description = 'Real Estate Property'

    # Add indexes on frequently searched/sorted fields
    property_code = fields.Char(
        string="Property Code",
        index=True,  # Database index for fast searches
        copy=False
    )
    state = fields.Selection([
        ('draft', 'Draft'),
        ('published', 'Published'),
        ('sold', 'Sold')
    ], string="Status", index=True, default='draft')

    create_date = fields.Datetime(index=True)  # For date-based queries

    # Efficient data retrieval
    def get_all_agent_names(self):
        # Single query with join
        properties = self.search([]).mapped('agent_id.name')
        return list(set(properties))  # Remove duplicates

    def get_property_summary(self):
        # Count without loading records
        return self.search_count([])
```

```
# Optimized computed field
@api.depends('offer_ids.price')
def _compute_best_offer(self):
    if not self:
        return

    # Single query to get max prices for all properties
    query = """
        SELECT property_id, MAX(price) as max_price
        FROM estate_property_offer
        WHERE property_id IN %s
        GROUP BY property_id
    """
    self.env.cr.execute(query, (tuple(self.ids),))
    price_map = dict(self.env.cr.fetchall())

    for record in self:
        record.best_offer = price_map.get(record.id, 0.0)

# Efficient search methods
@api.model
def search_published_properties(self, limit=None):
    """Optimized search for published properties"""
    domain = [('state', '=', 'published')]
    return self.search(domain, limit=limit, order='create_date desc')

# Use read() for specific fields when possible
def get_property_list_data(self):
    """Get minimal data for list views"""
    fields = ['name', 'asking_price', 'state', 'agent_id']
    return self.search([]).read(fields)
```

```
    # Batch operations
    def bulk_update_status(self, new_status):
        """Update multiple records at once"""
        # Single write operation instead of loop
        self.write({'state': new_status})

    # Use SQL for complex aggregations
    def get_sales_statistics(self):
        """Get sales statistics using raw SQL"""
        query = """
            SELECT
                COUNT(*) as total_properties,
                AVG(asking_price) as avg_price,
                SUM(CASE WHEN state = 'sold' THEN 1 ELSE 0 END) as sold_count
            FROM estate_property
            WHERE active = true
        """
        self.env.cr.execute(query)
        return dict(zip(['total', 'avg_price', 'sold'], self.env.cr.fetchone()))

# Server configuration for performance
# odoo.conf
[options]
# Optimize workers based on CPU cores
workers = 4
max_cron_threads = 2

# Memory limits
limit_memory_hard = 2684354560  # 2.5GB
limit_memory_soft = 2147483648  # 2GB
```

```
# Time limits
limit_time_cpu = 600    # 10 minutes CPU time
limit_time_real = 1200  # 20 minutes real time

# Database connection pooling
db_maxconn = 64

# Enable proxy mode for better performance
proxy_mode = True

# Log slow queries for optimization
log_level = info
log_sql = False  # Only enable for debugging
bash
# ✅ Performance monitoring and optimization

# 1. Monitor database performance
# Enable slow query logging in PostgreSQL
# postgresql.conf:
log_min_duration_statement = 1000  # Log queries > 1 second

# 2. Use Odoo profiling
./odoo-bin --addons-path=addons -d mydb --profile

# 3. Monitor memory usage
ps aux | grep odoo
htop  # Real-time monitoring

# 4. Database maintenance
# Regular VACUUM and ANALYZE
```

```
sudo -u postgres psql -d mydb -c "VACUUM ANALYZE;"

# 5. Optimize PostgreSQL configuration
# postgresql.conf optimizations:
shared_buffers = 256MB
effective_cache_size = 1GB
work_mem = 4MB
maintenance_work_mem = 64MB

# 6. Monitor with external tools
# Install and use tools like:
# - pg_stat_statements for PostgreSQL
# - New Relic or similar APM tools
# - Custom logging for application metrics
```

Conclusion

This comprehensive guide covers the top 50 most common errors encountered in Odoo 18 app development. Each error includes:

- **Clear problem identification** with symptoms
- **Wrong code examples** showing what not to do
- **Correct solutions** with proper implementation
- **Additional tips** and best practices

Key Takeaways:

1. **Always follow Odoo conventions** for naming, structure, and patterns
2. **Test incrementally** after each change
3. **Use proper field types** and relationships
4. **Implement security** from the beginning
5. **Monitor performance** and optimize as needed
6. **Keep dependencies updated** and properly declared
7. **Use version control** and backup regularly
8. **Read the official documentation** for the latest changes

Quick Reference for Prevention:

- ✅ **Module Structure**: Proper __manifest__.py, __init__.py, directory structure
- ✅ **Models**: Correct _name, field types, relationships, constraints
- ✅ **Views**: Valid XML, correct field references, proper widget usage
- ✅ **Security**: Access rights CSV, groups definition, field-level security
- ✅ **Server**: Proper configuration, database setup, performance optimization

For the most up-to-date information, always refer to the official Odoo 18.0 documentation.