Code Changes in Odoo 18

Odoo 18 has introduced a plethora of new features and improvements, not just in functionality but also in its code structure and syntax. For developers, understanding these changes is essential to seamlessly transition from older versions like Odoo 16. This article highlights the critical updates to Odoo 18's code syntax and provides guidance on adapting your modules.

1. XML Tag Changes: From tree to list

One of the most prominent changes in Odoo 18 is the renaming of the tree tag to list. This change is straightforward but affects all views that previously utilized the tree tag.

Before:

```
<tree>
  <field name="name"/>
</tree>
```

After:

```
<list>
  <field name="name"/>
</list>
```

This change enhances consistency and clarity in XML syntax.

2. attrs and states Attribute Simplifications

Odoo 18 simplifies the use of conditional attributes by replacing attrs and states with direct attributes.

Examples:

a. One condition

Before:

```
<field name="shift_id" attrs="{'invisible': [('shift_schedule', '=', [])]}"/>
```

b. Two Condition (with OR)

Before:

```
<field name="department_id" attrs="{'invisible': ['|', ('state', '=', 'done'), ('type', '=', 'internal')]}"/>
```

After:

```
<field name="department_id" invisible="state == 'done' or type == 'internal'"/>
```

c. Two conditions (with AND)

Before:

```
<field name="job_position" attrs="{'readonly': [('state', '=', 'approved'), ('user_id', '!=', user.id)]}"/>
```

After:

```
<field name="job_position" readonly="state == 'approved' and user_id != user.id"/>
```

Similarly, the states attribute is replaced by conditions like:

Before:

```
<button string="Submit" states="draft"/>
```

After:

```
<button string="Submit" invisible="state != 'draft'"/>
```

These changes make the XML cleaner and easier to maintain.

3. Widget Updates: The daterange Widget

The daterange widget has been updated for better functionality.

Before:

```
<div>
  <field name="start_date" widget="daterange" options="{'related_end_date': 'end_date'}"/>
  <field name="end_date" widget="daterange" options="{'related_start_date': 'start_date'}"/>
</div>
```

After:

```
<div>
  <field name="start_date" widget="daterange" options="{'end_date_field': 'end_date'}"/>
</div>
```

This reduces redundancy and simplifies the options configuration.

4. Chatter Simplification

Odoo 18 introduces a new way to define the chatter, eliminating the need for verbose XML.

Before:

```
<div class="oe_chatter">
  <field name="message_follower_ids" widget="mail_followers"/>
  <field name="activity_ids" widget="mail_activity"/>
  <field name="message_ids" widget="mail_thread"/>
</div>
```

After:

```
<chatter/>
```

For additional customization, you can use:

```
<chatter reload_on_follower="True"/>
```

5. Removal of Not Number Call and Doall in Scheduled Actions

Odoo 18 has deprecated the use of Not Number Call and Doall in scheduled actions (cron jobs). Ensure your custom actions align with these updates by refactoring your code.

6. States in Field Models Removed

Odoo 18 has removed the states attribute in Python field definitions. Instead, the logic should be handled directly in the UI or through other means.

Before:

```
date = fields.Date(
        string='Date',
        index=True,
        compute='_compute_date', store=True, required=True, readonly=False,
        precompu
        states={'posted': [('readonly', True)], 'cancel': [('readonly', True)]},
        copy=False,
        tracking=True,)
```

After:

```
date = fields.Date(
        string='Date',
        index=True,
        compute='_compute_date', store=True, required=True, readonly=False,
        precompu
        copy=False,
        tracking=True,)
```

7. Structural Changes in res.config XML

The structure of res.config.settings has been simplified in XML. Settings are now grouped into blocks with app and block tags.

Before:

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
    <data>
        <record id="res_config_settings_view_inherit_example" model="ir.ui.view"
            <field name="name">res.config.settings.view.form.inherit.example</fi
            <field name="model">res.config.settings</field>
            <field name="inherit_id" ref="base.res_config_settings_view_form"/>
            <field name="arch" type="xml">
                <xpath expr="//form" position="inside">
                    <div class="app_settings_block" data-
                    string="application_sett
                        <h2>Example Settings</h2>
                        <div class="row mt16 o_settings_container">
                        <label for="example_setting" string="Example
                        Setting"
```

```
                                        </div>
                                        <div class="row mt16 o_settings_container"
                                        name="example_s
                                        <field class="ml-4" name="example_setting"/>
                                        </div>
                                        <div class="row mt16 o_settings_container">
                                        <div class="text-muted ml-4">
                                        Description for the example setting.
                                        </div>
                                    </div>
                                    <div class="app_settings_block" data-
                                    string="Attendances" stri
                                </xpath>
                            </field>
                        </record>
                    </data>
</odoo>
```

After:

```
<record model="ir.ui.view" id="res_config_settings_view">
        <field name="arch" type="xml">
                <xpath expr="//form" position="inside">
                        <app string="Application Settings">
                                <block title="Example Settings">
                                        <setting string="Example Setting" help="Description for
                                        the
                                        <field name="example_setting"/>
                                        </setting>
                                </block>
                        </app>
                </xpath>
        </field>
</record>
```

Explanation:

From to <app> : The app_settings_block is replaced with the app tag to group settings logically.

From <h2> to <block> titles: Headers defined using <h2> are now incorporated directly as block titles.

From nested div containers to setting tags: Fields and their descriptions are encapsulated within setting tags, streamlining the layout.

Addition of help attributes: Inline descriptions are now provided using the help attribute within setting tags.

Conclusion

Migrating to Odoo 18 involves adapting to these syntactic changes to ensure compatibility and leverage the new features. These updates not only improve code readability but also align with modern development practices. Stay tuned for more insights into Odoo development, and don't forget to explore the official documentation for a deeper dive into Odoo 18.

Happy coding!