

TECHNICAL GUIDELINES

25-03-2020

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

Giới thiệu	3
Quy ước trong quá trình làm việc	3
Git & Github	3
Odoo Coding	4
Quy ước về module	4
Số phiên bản	5
Migrations	5
Directories	6
Tên file	6
Installation hooks	6
Cấu trúc đầy đủ	7
External Dependencies	8
XML File	9
Format	9
Views	9
QWeb	9
Menu	9
Đặt tên xml_id	10
Data Records	10
Security, View and Action	10
Inherited XML	11
Demo Records	12
Filter & Group	12
Python	12
Javascript	16
CSS	16
Quy ước khác	16
Quy chế về việc xử lý vi phạm	16

1. Giới thiệu

Đây là tài liệu miêu tả các quy ước coding, comment, quy trình kiểm soát công việc cũng như chất lượng sản phẩm,... áp dụng cho toàn bộ đội ngũ kỹ thuật trên toàn công ty.

Đồng thời tài liệu này cũng ghi rõ mức phạt (quy ra tiền) đối với từng trường hợp vi phạm cụ thể. Số tiền thu được sẽ được đưa vào quỹ vui chơi giải trí của công ty.

2. Quy ước trong quá trình làm việc

- 2.1. Luôn luôn online trên hệ thống viindoo.com trong giờ làm việc.
- 2.2. Thường xuyên làm việc cộng tác trên các hệ thống viindoo.com, github.com/Viindoo. Những thắc mắc hoặc câu hỏi phức tạp phải tạo issue hoặc comment trực tiếp vào PR. Việc này vừa để phục vụ việc truy vết lịch sử vừa tránh làm phiền người khác.
- 2.3. Trong quá trình nâng cấp một module từ phiên bản thấp lên cao. Nếu muốn thay đổi, cải tiến tính năng thì phải hỏi ý kiến của cấp phó phòng trở lên. Nếu đơn giản có thể hỏi trực tiếp, nếu phức tạp thì phải tạo issue và tag những người liên quan.
- 2.4. Khi ai đó mention đến mình trên hệ thống Viindoo + Github, đặc biệt người đó là cấp trên của mình thì trong vòng 4 giờ làm việc phải có reply cho thông tin đó.

3. Git & Github

- 3.1. Nội dung commit phải tham chiếu tới issue nếu có.
- 3.2. Nội dung commit phải chứa close #xxx (xxx là id của issue) để tự động close issue **nếu cần**.
- 3.3. Commit phải có nội dung rõ ràng nhưng ngắn gọn và xúc tích.
- 3.4. Một Pull Request (PR) tương ứng với một tính năng hoặc fix cho 1 lỗi. Chia càng nhỏ càng tốt. Để có thể ưu merge trước, tránh phải chờ hoàn thành tất cả mới merge.
- 3.5. Đối với người review phải thực hiện hoặc yêu cầu tác giả thực hiện rebase (nếu có commits behind) trước khi merge.
- 3.6. Một PR nếu có commits behind thì phải thực hiện rebase trước khi request review.
- 3.7. Tiền tố của nội dung commit phải tuân thủ theo quy ước sau:
 - 3.7.1. **[WIP]** = work in progress, sử dụng cho các commit mà đang ở trong quá trình thực hiện (chưa hoàn thiện tính năng). Rất phù hợp trong các công việc dài ngày mà vẫn phải commit hàng ngày mặc dù chưa hoàn thiện. **Lưu ý: nếu đang làm dở, chưa cần review thì bắt buộc phải để WIP**

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

- 3.7.2. **[IMP]** = improvements, sử dụng trong trường hợp commit cải tiến
- 3.7.3. **[FIX]** = Sửa lỗi, sử dụng trong trường hợp commit để sửa lỗi
- 3.7.4. **[MIG]** = migration, sử dụng trong trường hợp commit để migrate lên phiên bản Odoo cao hơn
- 3.7.5. **[I18N]**: cho các vấn đề liên quan đến translation
- 3.7.6. **[MERGE]**: cho commit để trộn 1 hay nhiều commit từ nhánh khác sang
- 3.7.7. **[UPG]**: nâng cấp module lên phiên bản Odoo sau
- 3.7.8. **[REM]**: xóa module
- 3.7.9. **[REN]**: đổi tên module
- 3.7.10. **[ADD]**: bổ sung tính năng, bổ sung module mới
- 3.7.11. **[MISC]**: các vấn đề khác
- 3.8. Tiền tố cho title của PR cũng tương tự như mục 3.7 ở trên chỉ khác là bổ sung thêm phiên bản của Odoo và tên module nếu PR đó chỉ liên quan đến một module, ví dụ: **[UPG] [13.0]** module_name....
- 3.9. Nội dung của commit phải hoàn toàn bằng **tiếng Anh**.
- 3.10. Quy ước về tên branch
 - 3.10.1. **v<odoo_version>_fix_<ten_module>**: nếu là fix bug cho một module
 - 3.10.2. **v<odoo_version>_upg_<ten_module>**: nếu là nâng cấp phiên bản (odoo version) cho một module
 - 3.10.3. **v<odoo_version>_add_<ten_module>**: nếu là thêm mới module
 - 3.10.4. **v<odoo_version>_rem_<ten_module>**: nếu là remove module
 - 3.10.5. **v<odoo_version>_imp_<ten_module>**: nếu là cải tiến module
 - 3.10.6. **v<odoo_version>_<ting_nang>**: nếu là một bộ tính năng nào đó.
Lưu ý **<ting_nang>** cần ngắn gọn và súc tích
- 3.11. Cấu hình đúng email và name trước khi commit
- 3.12. Thực hiện yêu cầu review nếu PR đã làm xong và cần review. Việc này sẽ giảm thời gian cho reviewer phải đặt câu hỏi kiểu như "làm xong chưa?"
- 3.13. PR phải có miêu tả rõ ràng hoặc phải gắn tới link mà có miêu tả rõ ràng trong đó.
- 3.14. Các file không được để hiển thị thông báo no newline trên github
- 3.15. KHÔNG được gán các nhãn của runbot một cách THỦ CÔNG vào issue hoặc PR, ví dụ như: running, failed, building,... Việc này phải để runbot thực hiện.
- 3.16. Trong quá trình review 1 PR, nếu PR đó là 1 hoặc 1 cụm tính năng phức tạp thì khi xử lý các comments của người review ta nên tạo commit mới. Mục đích của việc này nhằm giúp người review dễ check các thay đổi, tránh phải review lại cả file. Đặc biệt là với file dài.
- 3.17. Khi có yêu cầu quay clip trên PR thì phải upload trực tiếp lên github thay vì lưu vào các chỗ cá nhân như youtube, google drive,...

- 3.18. Khi 1 PR bị phụ thuộc vào các PR khác thì trong mô tả phải thêm đường dẫn (link) đến các PR phụ thuộc, đồng thời gắn labels (Blocked by dependencies) vào PR

4. Odoo Coding

4.1. Quy ước về module

- 4.1.1. Tên module: viết thường, không dấu, ngăn cách bởi `_`. Phải bắt đầu bằng `to_`. Với những module mới thì tiền tố sẽ là **viin_**
- 4.1.2. Khi kế thừa và mở rộng một module khác (ví dụ module `hr_expense`) thì tiền tố phải là **viin_hr_expense_****xxx**.
- 4.1.3. Khi tạo một module bản địa hóa thì tiền tố phải có dạng **viin_l10n_cc_****xxx**, trong đó `cc` là mã quốc gia (country code). Ví dụ: `to_l10n_vn_employee_advance`.
- 4.1.4. File `__manifest__.py` phải bao gồm đầy đủ các thông tin sau:
 - **name**: tên module bằng tiếng Anh
 - **name_vi_VN**: tên module bằng tiếng Việt
 - **version**: phiên bản của module. Mặc định là **0.1.0**
 - **author**: Khi có thay đổi sẽ cập nhật vào tài liệu này. Những module mới sẽ là "*Viindoo*", những module cũ là "*T.V.T Marine Automation (aka TVTMA), Viindoo*"
 - **website**: <https://viindoo.com>
 - **live_test_url**: <https://vxxdemo-int.erponline.vn>, trong đó **xx** là phiên bản Odoo ví dụ: 12, 13, 14,...
 - **support**: apps.support@viindoo.com
 - **summary**: miêu tả ngắn gọn bằng tiếng Anh
 - **summary_vi_VN**: miêu tả ngắn gọn bằng tiếng Việt.
 - **category**: category của module
 - **sequence**: trình tự của module
 - **description**: miêu tả chi tiết của module bằng tiếng Anh, tuyệt đối tuân thủ cú pháp Markdown. Trong nội dung miêu tả phải bao gồm thông tin Editions Supported: Community Edition và Enterprise Edition
 - **description_vi_VN**: miêu tả chi tiết của module bằng tiếng Việt, tuyệt đối tuân thủ cú pháp của Markdown. Trong nội dung miêu tả phải bao gồm thông tin Editions Supported: Community Edition và Enterprise Edition
 - **installable**: mặc định là True, nếu là module chưa sẵn sàng để sử dụng thì là False.
 - **application**: là True nếu module là app, không thì là False
 - **auto_install**: thông thường với những module dạng cầu nối cho một số modules khác thì phải thiết lập là True, không thì là False.

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

- **price:** mặc định là 99.9
- **currency:** mặc định là EUR
- **license:** mặc định là OPL-1

- 4.1.5. Tuyệt đối không đổi tên module (tên kỹ thuật) sau khi đã thiết lập installable: True. Việc đổi tên module (nếu có) chỉ được thực hiện ở các phiên bản nâng cấp sau đó. Khi đổi tên module, trong file manifest phải có thêm cặp key => val: **'old_technical_name': 'xxx'**, trong đó **xxx** là tên module cũ
- 4.1.6. Summary của module không được dài quá 158 ký tự. Lý do: summary không dùng để mô tả chi tiết tính năng và đặc biệt nó được dùng làm meta title trên website app store nên nếu dài quá sẽ bị hạ rank

4.2. Số phiên bản

Đối với module mới, phiên bản đầu tiên sẽ là **0.1.0**. Khi nào cần migration thì mới tăng lên, ví dụ: 0.1.1, 0.1.2,...

4.3. Migrations

Khi module có thay liên quan đến cấu trúc database hoặc làm ảnh hưởng đến dữ liệu đang có sẵn thì cần phải viết migration script để nâng cấp phiên bản module. Khi viết migration cần tách thành các hàm nhỏ và được gọi trong phương thức migrate.

4.4. Directories

Một module được tổ chức với các thư mục như sau:

- **controllers/:** chức các controllers (http routes)
- **data/:** data xml
- **demo/:** demo xml
- **examples/:** external files lib/,...
- **models/:** model definitions
- **reports/:** chứa các model báo cáo (BI/analysis), webkit report templates
- **static/:** bao gồm web assets, thư mục css/, js/, img/,...
- **templates/:** chứa các web templates
- **views/:** bao gồm các views và các mẫu in QWeb report
- **wizards/:** wizard model and views

4.5. Tên file

Đối với model, views và data thì cần chia nhỏ file theo từng model. Một số trường hợp ngoại lệ đối với one2many thì có thể viết cùng file views với main

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

model, nhưng file python vẫn phải tách riêng. Ví dụ views của *sale.order.line* có thể được viết cùng với views của *sale.order*.

- `models/<main_model>.py`
- `data/<main_model>_data.xml`
- `demo/<main_model>_demo.xml`
- `templates/<main_model>_templates.xml`
- `views/<main_model>_views/xml`

Đối với controller, nếu chỉ có một file thì nên đặt tên là `main.py`. Nếu nhiều hơn thì có thể chia thành nhiều file.

Đối với static, tên file có quy ước là `<module_name>.ext` (ví dụ `static/im_chat.js`, `static/css/im_chat.css`, `static/xml/im_chat.xml`,...)

Đối với security, access right và rules thì sử dụng các files chính sau:

- `ir.model.access.csv`: định nghĩa quyền truy cập trên các models
- `module_security.xml`: các groups nằm trên cùng và nằm trong 1 thẻ data riêng. Các record rules nằm trong 1 thẻ data khác và các rules của cùng một model phải nằm liên tiếp nhau không được lẫn lộn các rules của model khác.

4.6. Installation hooks

Khi viết hooks, cần chia nhỏ các tính năng thành các hàm và gọi nó ở trong phương thức hook.

4.7. Cấu trúc đầy đủ

Cấu trúc đầy đủ của một module trông sẽ như sau:

```
<my_module_name>/
|-- __init__.py
|-- __manifest__.py
|-- controllers/
|   |-- __init__.py
|   `-- main.py
|-- data/
|   `-- <main_model>.xml
|-- demo/
|   `-- <inherited_model>.xml
|-- migrations/
|   `-- 12.0.x.y.z/
|       |-- pre_migration.py
|       `-- post_migration.py
|-- models/
|   |-- __init__.py
|   |-- <main_model>.py
|   `-- <inherited_model>.py
```

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

```
|-- report/
|   |-- __init__.py
|   |-- report.xml
|   |-- <bi_reporting_model>.py
|   |-- report_<rml_report_name>.xml
|   |-- report_<rml_report_name>.py
|   |-- <webkit_report_name>.mako
|-- security/
|   |-- ir.model.access.csv
|   |-- module_security.xml
|-- static/
|   |-- img/
|   |   |-- my_little_kitten.png
|   |   |-- troll.jpg
|   |-- lib/
|   |   |-- external_lib/
|   |-- src/
|   |   |-- js/
|   |   |   |-- <my_module_name>.js
|   |   |-- css/
|   |   |   |-- <my_module_name>.css
|   |   |-- less/
|   |   |   |-- <my_module_name>.less
|   |   |-- xml/
|   |   |   |-- <my_module_name>.xml
|-- tests/
|   |-- __init__.py
|   |-- <test_file>.py
|   |-- <test_file>.yml
|-- views/
|   |-- <main_model>_views.xml
|   |-- <inherited_main_model>_views.xml
|   |-- report_<qweb_report>.xml
|-- templates/
|   |-- <main_model>.xml
|   |-- <inherited_main_model>.xml
|-- wizards/
|   |-- __init__.py
|   |-- <wizard_model>.py
|   |-- <wizard_model>.xml
|-- examples/
|   |-- my_example.csv
```

Tên file chỉ nên dùng [a-z0-9]

4.8. External Dependencies

Nếu trong module có sử dụng thư viện python bên ngoài thì phải thêm thông tin external_dependencies vào trong __manifest__.py và bổ sung thêm vào file requirements.txt (tạo mới file này nếu chưa có) của repository hiện hành. Đồng

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

thời, cần thông báo với phòng Hệ thống Nội bộ để cài thư viện lên server và phải confirm là đã cài đầy đủ mới được merge.

```
{
  'name': 'Example Module',
  'external_dependencies': {
    'bin': [
      'external_dependency_binary_1',
      'external_dependency_binary_2',
    ],
    'python': [
      'external_dependency_python_1',
      'external_dependency_python_2',
    ],
  },
  'installable': True,
}
```

4.9. XML File

4.9.1. Format

Quy ước về record trong XML:

- Thuộc tính id phải đặt trước model
- Đối với field, thuộc tính name phải được đặt đầu tiên xong mới đến các thuộc tính khác như widget, options,...
- Không sử dụng **module_name.view_id**.

```
<record id="view_id" model="ir.ui.view">
  <field name="name">view.name</field>
  <field name="model">object_name</field>
  <field name="priority" eval="16"/>
  <field name="arch" type="xml">
    <tree>
      <field name="my_field_1"/>
      <field name="my_field_2" string="My Label"
widget="statusbar" statusbar_visible="draft,sent,progress,done"
statusbar_colors='{ "invoice_except": "red", "waiting_date": "blue"
}' />
    </tree>
  </field>
</record>
```

4.9.2. QWeb

- t-*-options (t-field-options, t-esc-options và t-row-options) không được sử dụng từ phiên bản 10 trở lên, vì chúng đã bị xóa sau phiên bản 10:

<https://www.odoo.com/documentation/10.0/reference/views.html#lists>

4.9.3. Menu

- Các menu root mà không gán tới action của model nào thì nằm trong một file xml riêng có tên **root_menu.xml**.
- Các menu liên quan đến model nào thì để trong views tương ứng của model đó.
- Các menu app thì bắt buộc phải có web_icon.

4.9.4. Đặt tên xml_id

Data Records

Sử dụng format **<model_name>_<record_name>**

```
<record id="res_users_important_person" model="res.users">
    ...
</record>
```

Security, View and Action

- Đối với menu: **<model_name>_menu**
- Đối với view: **<model_name>_view_<view_type>**, trong đó view_type có thể là tree, form, kanban, search,...
- Đối với action: action chính sẽ có dạng **<model_name>_action**. Trong trường hợp có nhiều actions cho một model thì các action chi tiết sẽ có thêm hậu tố **_<details>**
- Đối với group: **<module_name>_group_<group_name>**, trong đó group_name là tên của group, thường sẽ là user hoặc administrator
- Đối với rule: **<model_name>_rule_<concerned_group>** trong đó concerned_group là tên viết tắt của nhóm quyền mà rule này nó tác động vào. Ví dụ đối với nhóm group_user thì concerned_group nên là 'user', đối với nhóm public user thì concerned_group nên là 'public', đối với multi-company rule thì concerned_group nên là 'company',...

```
<!-- views and menus -->
<record id="model_name_menu" model="ir.ui.menu">
    ...
</record>

<record id="model_name_view_form" model="ir.ui.view">
    ...
</record>

<record id="model_name_view_kanban" model="ir.ui.view">
    ...
</record>

<!-- actions -->
<record id="model_name_action" model="ir.actions.act_window">
```

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

```
...
</record>

<record id="model_name_action_child_list" model="ir.actions.act_window">
  ...
</record>

<!-- security -->
<record id="module_name_group_user" model="res.groups">
  ...
</record>

<record id="model_name_rule_public" model="ir.rule">
  ...
</record>

<record id="model_name_rule_company" model="ir.rule">
  ...
```

Inherited XML

Trong một module chỉ có thể mở rộng 1 view 1 lần. Ví dụ: nếu muốn mở rộng form sản phẩm và thêm 2 trường field_1, field_2 thì chỉ nên tạo 1 view để hiển thị cả 2 trường này thay vì tạo ra 2 view (mỗi view hiển thị 1 trường).

Tuyệt đối không sử dụng <... position="replace"> nếu không có lý do đặc biệt. Việc này có thể gây ra lỗi 'Element ... cannot be located in parent view' từ view hoặc module khác. Nếu bắt buộc phải sử dụng thì phải comment giải thích rõ lý do tại sao phải làm vậy và phải set priority cao (khoảng > 100) cho view đó để tránh lỗi.

```
<record id="view_id" model="ir.ui.view">
  <field name="name">view.name</field>
  <field name="model">object_name</field>
  <field name="priority">110</field> <!--Priority greater than 100-->
  <field name="arch" type="xml">
    <!-- It is necessary because...-->
    <xpath expr="//field[@name='my_field_1']" position="replace"/>
  </field>
</record>
```

Xml id của các view kế thừa nên sử dụng cùng với ID của view gốc. Việc này sẽ giúp tìm kiếm tất cả các view kế thừa một cách dễ dàng.

Demo Records

Luôn luôn phải có hậu tố demo ở cuối mỗi demo record để phân biệt với các record thông thường

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

```
<record id="res_users_not_a_real_user_demo" model="res.users">
    ...
</record>
```

4.9.5. Filter & Group

- Đối với bộ lọc thì tên của bộ lọc có tiền tố là **ftt_**
- Đối với nhóm (group) thì tên của nhóm có tiền tố là **grp_**
- Khi thiết kế views phải luôn nghĩ đến filter và group. Ví dụ: nếu model hr.employee mà có trường department_id many2one tới hr.department thì phải có group by department, nếu có state thì phải có filter theo state.

4.10. Python

4.10.1. Imports

Thứ tự import như sau:

- Các thư viện tiêu chuẩn
- Các thư viện của bên thứ 3 mà chắc chắn sẽ có
- Các thư viện Odoo
- Import các modules của Odoo
- Local import
- Các thư viện của bên thứ 3 mà không chắc là sẽ có.

```
# 1: imports of python lib
import base64
import logging
import re
import time

# 2: import of known third party lib
import lxml

# 3: imports of odoo
import odoo
from odoo import api, fields, models # alphabetically ordered
from odoo.tools.safe_eval import safe_eval
from odoo.tools.translate import _

# 4: imports from odoo modules
from odoo.addons.website.models.website import slug
from odoo.addons.web.controllers.main import login_redirect

# 5: local imports
from . import utils

# 6: Import of unknown third party lib
_logger = logging.getLogger(__name__)
```

```
try:
    import external_dependency_python_N
except ImportError:
    _logger.debug('Cannot `import external_dependency_python_N`.')
```

4.10.2. Quy ước khi viết code python

- Sử dụng UpperCamelCase cho tên class, ví dụ: class AccountInvoice
- Luôn luôn đặt tên biến có ý nghĩa, thậm chí comment rõ ràng nếu cần
- Đối với các trường trong Odoo phải có String cho mọi trường hợp và nếu cảm thấy người khác đọc trường này không hiểu nó dùng để làm gì thì **bắt buộc** phải có help string.
- Đối với trường Many2one phải có `_id` ở cuối
- Đối với trường X2many phải có `_ids` ở cuối.
- Đối với trường related: nếu một trường related đến một trường có `required=True` thì phải `store=False` hoặc `readonly=False`
- Sử dụng gạch dưới và viết thường cho các tên biến
- Nếu là record hoặc recordsets thì không sử dụng `_id` hoặc `_ids`
- Sử dụng map, filter, sum,... để code dễ đọc và hiệu quả hơn.
- Docstring của method nên giải thích mục đích và giá trị của function thay vì tóm tắt nội dung của code.
- Nếu một function quá dài hoặc chứa quá nhiều vòng lặp lồng nhau ở bên trong thì cần phải xem xét tái cấu trúc thành các function nhỏ hơn.
- Khi lập recordset thì đặt tên biến là `r` hoặc `record` nếu là một instance của object hiện hành. Nếu không phải thì không đặt là `r` hoặc `record`.
- Luôn sử dụng `%` và `%(varname)` thay cho positional để dễ dàng cho việc đọc và dịch.
- Khi sử dụng hàm get của dictionary, bắt buộc phải có giá trị default để tránh lỗi nếu key không có trong dict.
- Không sử dụng tiền tố `_compute` hoặc `compute` nếu đó không phải là phương thức dùng để compute 1 trường.
- Sử dụng nháy đơn cho các string và key của dictionary, nháy kép cho các message.
 - Thuộc tính String của trường: nháy đơn
 - Thuộc tính help của trường: nháy kép
 - Các message: nháy kép
 - String trong dictionary, tuple, list: nháy đơn
 - Docstring: 3 nháy kép
- Tuyệt đối không bổ sung thêm argument bắt buộc vào các function/method của một phiên bản ổn định vì nó có thể break các module khác mà phụ thuộc vào module này, bao gồm cả module của chúng ta và module mà các khách hàng của module này phát triển thêm

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

4.10.3. No SQL Injection

KHÔNG BAO GIỜ sử dụng toán tử + chuỗi để truyền biến vào SQL query string.

```
# the following is very bad:
# - it's a SQL injection vulnerability
# - it's unreadable
# - it's not your job to format the list of ids
cr.execute('select distinct child_id from
account_account_consol_rel ' +
          'where parent_id in ('+', '.join(map(str, ids))+')')

# better
cr.execute('SELECT DISTINCT child_id '\
          'FROM account_account_consol_rel '\
          'WHERE parent_id IN %s',
          (tuple(ids),))
```

4.10.4. Do not bypass the ORM

Không được truy vấn trực tiếp vào database nếu như ORM có thể làm điều tương tự

```
# very very wrong
cr.execute('select id from auction_lots where auction_id in ('
+
          ', '.join(map(str, ids)) + ') and state=%s and
obj_price>0',
          ('draft',))
auction_lots_ids = [x[0] for x in cr.fetchall()]

# no injection, but still wrong
cr.execute('select id from auction_lots where auction_id in %s
'
          'and state=%s and obj_price>0',
          (tuple(ids), 'draft',))
auction_lots_ids = [x[0] for x in cr.fetchall()]

# better
auction_lots_ids = self.search(cr, uid, [
    ('auction_id', 'in', ids),
    ('state', '=', 'draft'),
    ('obj_price', '>', 0),
])
```

4.10.5. Models

- Models names:
 - Viết thường và ngăn cách bởi dấu chấm. Ví dụ: account.invoice

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

- Tên model ở dạng số ít, ví dụ account.invoice thay vì account.invoices
- Method:
 - Compute Field: có dạng `_compute_<field_name>`
 - Inverse method: có dạng `_inverse_<field_name>`
 - Search method: có dạng `_search_<field_name>`
 - Default method: có dạng `_default_<field_name>`
 - Onchange method: có dạng `_onchange_<field_name>`
 - Constraint method: có dạng `_check_<constraint_name>`
 - Action method: có tiền tố `action_`. Decorator của nó là `@api.multi` (chỉ áp dụng cho phiên bản 12 trở về trước), nếu muốn sử dụng với 1 record thì phải có `self.ensure_one()` ở đầu method.
- Thứ tự khai báo trong model:
 - Private attributes: `_name`, `_description`, `_inherit`. **Lưu ý:** với model mới tạo ra, luôn luôn phải có `_description`.
 - `_default_<field_name>`
 - Khai báo các Fields
 - `_sql_constrains`
 - `_default_get`
 - Các phương thức search và compute
 - Các phương thức constrains và onchange
 - Các phương thức CRUD (ORM overrides)
 - Các phương thức action
 - Cuối cùng, các phương thức nghiệp vụ khác

```
class Event(models.Model):
    # Private attributes
    _name = 'event.event'
    _description = 'Event'

    # Fields declaration
    name = fields.Char(default=lambda self:
self._default_name())
    seats_reserved = fields.Integer(
        oldname='register_current',
        string='Reserved Seats',
        store=True,
        readonly=True,
        compute='_compute_seats',
    )
    seats_available = fields.Integer(
        oldname='register_avail',
        string='Available Seats',
        store=True,
        readonly=True,
        compute='_compute_seats',
    )
```

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

```
price = fields.Integer(string='Price')

# Default methods
def _default_name(self):
    ...

# compute and search fields, in the same order that fields
declaration
@api.multi
@api.depends('seats_max', 'registration_ids.state')
def _compute_seats(self):
    ...

# Constraints and onchange
@api.constrains('seats_max', 'seats_available')
def _check_seats_limit(self):
    ...

@api.onchange('date_begin')
def _onchange_date_begin(self):
    ...

# CRUD methods
def create(self):
    ...

# Action methods
@api.multi
def action_validate(self):
    self.ensure_one()
    ...

# Business methods
def mail_user_confirm(self):
    ...
```

- 4.10.6. Không sử dụng onchange để validate dữ liệu. Nếu có thì onchange chỉ warning mà thôi, thay vào đó phải dùng constrains để validate dữ liệu. Lý do: onchange chỉ chạy trên view và vẫn lưu lại được mặc dù không thỏa mãn điều kiện.
- 4.10.7. Tuyệt đối không sử dụng **AsIs** khi chưa xử lý dữ liệu đầu vào. Lý do: https://github.com/Viindoo/tvtmaaddons/pull/4243#discussion_r752137012
- 4.10.8. Khi viết automation test (hoặc tương tự) phải sử dụng email có đuôi là **@example.viindoo.com**

4.11. Javascript

- Không bao giờ thêm một thư viện Javascript đã minified
- Sử dụng UpperCamelCase để khai báo class
- Hạn chế sử dụng javascript trừ khi bắt buộc

4.12. CSS

- Không viết thêm css nếu bootstrap có thể giải quyết

4.13. Môi trường phát triển

- 4.13.1. Bắt buộc phải sử dụng hệ điều hành Ubuntu đối với máy tính được công ty cấp. Trừ một số trường hợp được CTO phê duyệt.
 - 4.13.2. Bắt buộc phải sử dụng Eclipse để dev Odoo trừ một số trường hợp được CTO phê duyệt.
 - 4.13.3. Trong quá trình code Odoo, bắt buộc phải add cùng lúc tất cả các repositories sau (được gọi là bộ repo. tiêu chuẩn của Viindoo):
 - <https://github.com/Viindoo/odoo>
 - <https://github.com/Viindoo/tvtmaaddons>
 - <https://github.com/Viindoo/erponline-enterprise>
 - <https://github.com/Viindoo/web> . Từ 14 trở đi không cần dùng repository này nữa
 - <https://github.com/Viindoo/branding>
 - https://github.com/Viindoo/backend_theme
 - https://github.com/Viindoo/odoo-web_gantt
 - 4.13.4. Phải sử dụng **python venv** cho từng phiên bản Odoo, cụ thể:
 - python 2.7 cho các phiên bản Odoo <= 10.0
 - python 3.6 cho Odoo 11.0
 - python 3.7 cho Odoo 12.0 và 13.0
 - python 3.8 cho Odoo 14.0 và 15.0
 - python 3.10 và PostgreSQL 14 cho Odoo 16.0
 - 4.13.5. Đường dẫn **python venv** trên local phải được đặt ở thư mục home, cụ thể: **~/python-venv/3.7 (các phiên bản python tương ứng)/odoo12 (các phiên bản odoo tương ứng)**
 - 4.13.6. Môi trường IDE luôn luôn phải nhìn thấy Odoo log
 - 4.13.7. Luôn luôn phải để ý log trong quá trình dev. Nếu có **error** hoặc **warning** phải xử hoặc thông báo cho người liên quan, người chịu trách nhiệm biết để xử lý. Mục này trước mắt chưa áp dụng hình thức phạt, sẽ xem xét nếu thường xuyên vi phạm.
 - 4.13.8. Khi chạy Odoo bắt buộc phải có tham số **--load=base,web,to_base**
- 4.14. Quy ước về file dịch i18n
- 4.14.1. Khi nâng cấp lên phiên bản mới phải xuất lại file dịch
 - 4.14.2. Thông tin Creation-Date và Revision-Date khi lên phiên bản mới. Còn sau đó nếu có IMP hay FIX thì KHÔNG thay đổi 2 thông tin này. Lý do: giảm thiểu conflict trong quá trình forward

5. Quy ước khác

- 5.1. Đối với coder: test trước khi push commit. Ít nhất phải tránh được các bug cơ bản. Tránh code chạy, không test mà vẫn push.
- 5.2. Tất cả các module tạo mới phải viết testcase.

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

- 5.3. Đối với reviewer: ít nhất phải test các case cơ bản trước khi approve. Ít nhất phải tránh được bug ngay khi cài module, tránh review chạy mà vẫn approve.
- 5.4. Khi tiếp nhận task/issue, kỹ thuật phải phản biện và check kỹ xem có hợp lý hay không rồi mới bắt tay vào code. Nếu PR đẩy lên mà bị cấp quản lý (phó phòng trở lên) reject vì lý do không phản biện và kiểm tra kỹ thì cả người làm PR lẫn reviewer đều bị phạt.
- 5.5. Fix bug hoặc cải tiến mà cải tiến đó làm thay đổi logic tính toán thì phải sửa hoặc bổ sung test case cho automation test. Trường hợp không thể sửa hoặc bổ sung test thì phải nêu rõ lý do cho người review biết.
- 5.6. Không fix code trực tiếp vào bản fork Odoo của Viindoo <https://github.com/Viindoo/odoo>. Trừ trường hợp cải tiến/fix file dịch hoặc có sự chỉ định của cấp phó phòng trở lên.
- 5.7. Không để xảy ra warning về cú pháp rst
- 5.8. Quy ước về nhãn và string:
 - 5.8.1. Đối với Label, menu

6. Quy chế về việc xử lý vi phạm

Vi Phạm lần đầu: Nhắc nhở

Vi phạm lần thứ 2 trở đi: khiển trách và phạt vi phạm với các mức như sau:

(x1000 VNĐ / lần)

CODE	Nội dung	Số tiền				
		TGD	GD	Trưởng /Phó BP	Trưởng nhóm	Còn lại
V1	Vi phạm mục 2.1	500	400	300	200	100
V2	Vi phạm mục 2.2	500	400	300	200	100
V3	Vi phạm mục 2.3, 2.4	500	400	300	200	100
V4	Vi phạm mục 3.1	500	400	300	200	100
V5	Vi phạm mục 3.2	500	400	300	200	100
V6	Vi phạm mục 3.5	500	400	300	200	100
V7	Vi phạm một trong các mục 3.6, 3.7. 3.8	250	200	150	100	50
V8	Vi phạm mục 3.9	1000	800	600	400	200
V9	Vi phạm mục	500	400	300	200	100

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO*Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng*

	3.10, 3.11, 3.12, 3.13					
V10	Vi phạm mục 3.14	250	200	150	100	50
V11	Vi phạm mục 3.15	1000	800	600	400	200
V12	Vi phạm mục 3.16, 3.17, 3.18	250	200	150	100	50
V13	Vi phạm mục 3.1	500	400	300	200	100
V14	Vi phạm mục 4.1, 4.2	250	200	150	100	50
V15	Vi phạm mục 4.3	500	400	300	200	100
V16	Vi phạm mục 4.5, 4.6	250	200	150	100	50
V17	Vi phạm mục 4.8, 4.9	500	400	300	200	100
V18	Vi phạm mục 4.10.1	500	400	300	200	100
V19	Vi phạm mục 4.10.2	250	200	150	100	50
V20	Vi phạm mục 4.10.3	1000	800	600	400	200
V21	Vi phạm mục 4.10.4	500	400	300	200	100
V22	Vi phạm mục 4.10.5	250	200	150	100	50
V23	Vi phạm mục 4.10.6	250	200	150	100	50
V24	Vi phạm mục 4.10.7	1000	800	600	400	200
V25	Vi phạm mục 4.10.8	250	200	150	100	50
V26	Vi phạm mục 4.11	500	400	300	200	100

CÔNG TY CỔ PHẦN CÔNG NGHỆ VIINDOO

Tầng 6, Tòa nhà Taiyo, số 97 Bạch Đằng, Hạ Lý, Hồng Bàng, Hải Phòng

V27	Vi phạm mục 4.13	250	200	150	100	50
V28	Vi phạm mục 4.14	250	200	150	100	50
V29	Vi phạm mục 5.1 5.4, 5.5, 5.6	250	200	150	100	50
V30	Vi phạm mục 5.7	500	400	300	200	100

Toàn bộ số tiền nộp phạt được dùng phục vụ cho việc liên hoan chung của anh em nhân viên trong công ty.