2019 odoo SERVICES

TfFC Using Studio

1. Introduction & Views

Julien Domken - Business Analyst

Contents taken from LNA's training.

0. Welcome

About this training - WIIFY

- Goal :
  - Show the potential
  - Give a general understanding >< Being exhaustive (and exhausting !)

To make sure you have the tools you'll need in your daily activities

TfFC Using Studio Agenda

1.	**Introduction & views**
2.	Automation & Access Rights
3.	Reports
4.	Create a new app

Introduction & Views

1.	Introduction to Studio
2.	Technical Concepts : Quick Refresher
3.	Technical Concept : Views
4.	Exercices

Introduction

Warning !

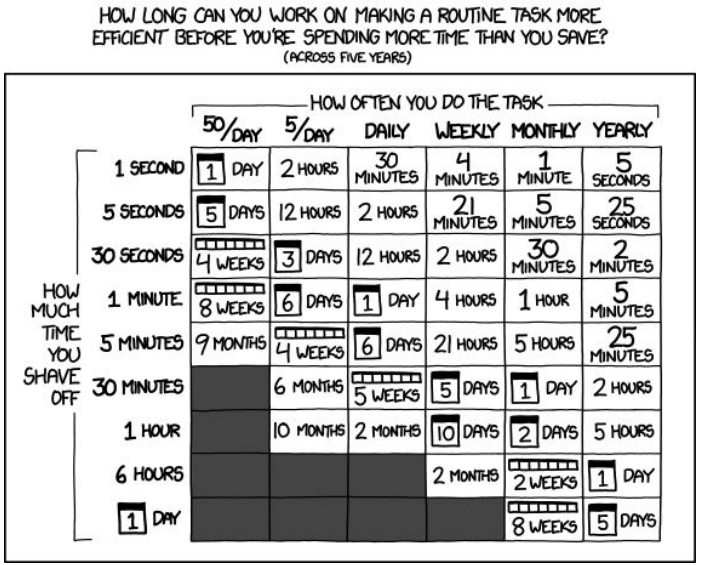
Elevator Pitch : Studio

- With vs Without Studio
- Test, Phase & Send to Production with only a few clicks (Studio Module)

A few words on PM

Introduction

- The “Quickstart” Methodology
- The “80/20” rule
- Educate your SPOC



Odoo Studio :

3 Pitfalls to avoid to make your project a success

- Reinventing the wheel
- Customization vs Development
- Not focusing on the added value & the Business Need

I - Reinventing the wheel

Take a look at what exists…

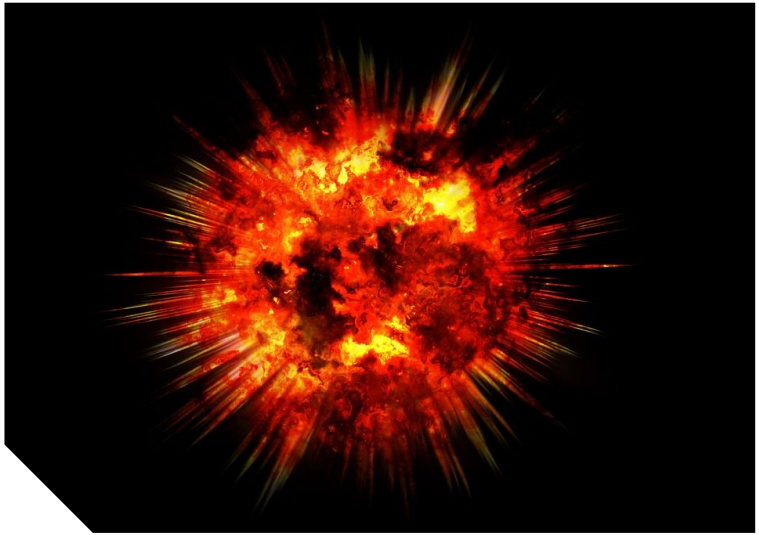
… More than 45 apps & 400 modules...

… and no need to reinvent them !

Ex : Invoicing & Sales Order

I - Two approaches, Same problem

The ‘Big Bang’



The ‘Creative’

II - Customization vs Development

“If a picture is worth a thousand words, sometimes a development is worth a thousand customizations !”

II - Customization vs Development

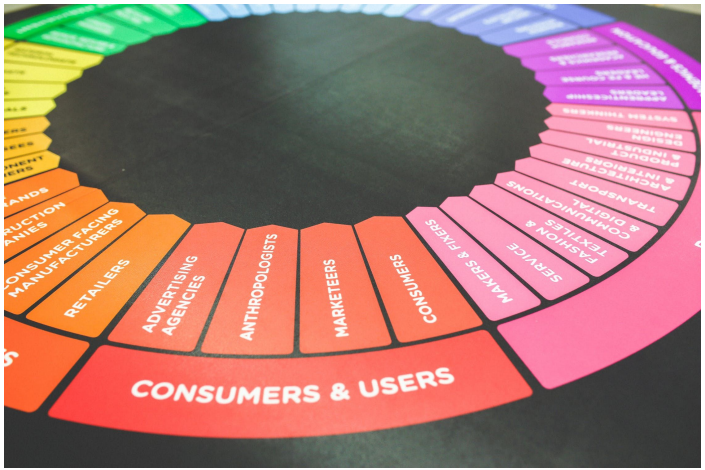
Interface Change VS Business flow

Standard Automation VS Complex Process

Adapting VS Creating

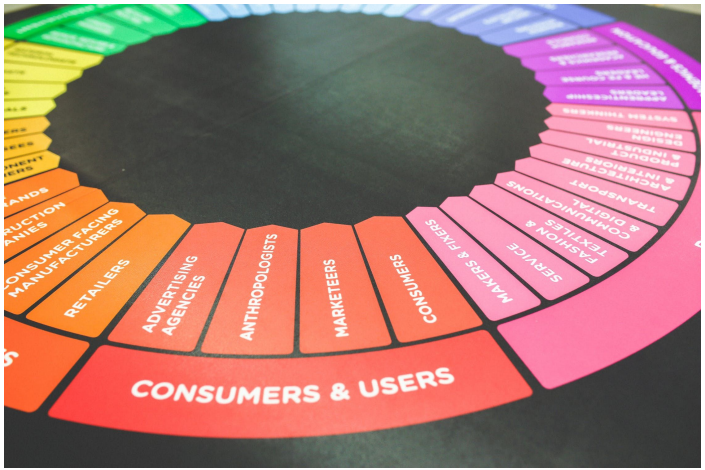
III - Not focusing on added-value

- Is the technical cost worth it ?
  - Cost : time, money, complexity
- Does it impact a lot of users ?
  - Role distribution
- Why?
  - Why? Why? Why?...



III - Not focusing on added-value

- Does it ease the onboarding ?
  - Not Replicating
- Does it impact the business ?
  - Think business not Odoo



IV - All in all...

-> Studio and its technical limits

-> The limit you should set yourself/your customer

Technical Concepts : Quick Refresher

PostgreSQL

DataBase server

ORM

(Object Relational Mapping)
Odoo Server

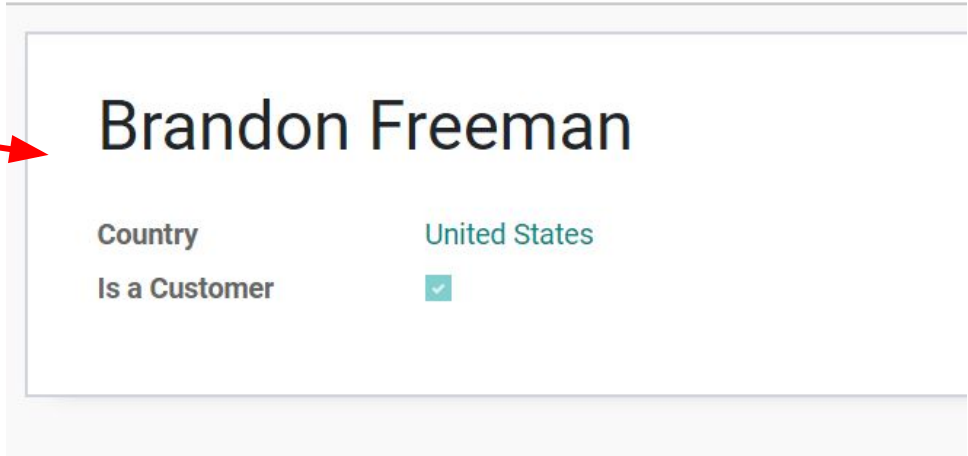
FrontEnd

Odoo Web

From a database to a UI

My Contacts (res.partner data table)		
name	country_id	customer
Brandon Freeman	United States	TRUE

```
<data>
  <xpath expr="//form" position="replace">
      <form>
          <sheet>
            <h1>
                <field name="name"/>
            </h1>
          <group>
              <field name="country_id"/>
              <field name="customer"/>
          </group>
          </sheet>
      </form>
  </xpath>
</data>
```



Odoo Database Structure:

From a database to a UI

PostgreSQL

DataBase server

ORM

(Object Relational Mapping)
Odoo Server

FrontEnd

Odoo Web

My Contacts (res.partner data table)		
name	country_id	customer
Brandon Freeman	United States	TRUE

```
<data>
  <xpath expr="//form" position="replace">
      <form>
          <sheet>
            <h1>
                <field name="name"/>
            </h1>
          <group>
              <field name="country_id"/>
              <field name="customer"/>
          </group>
          </sheet>
      </form>
  </xpath>
</data>
```

What is a database ?

Comparison with the Excel file

- Sheets are models
- Columns are fields
- Rows are records

id	name	city	email
base.res_partner_1	ASUSTeK	Taipei	asusteK@yourcompany.example.com
base.res_partner_address_27	Arthur Gomez	Taipei	
base.res_partner_address_12	James Miller	Taipei	
base.res_partner_address_2	Joseph Walters	Taipei	joseph.walters@asustek.com

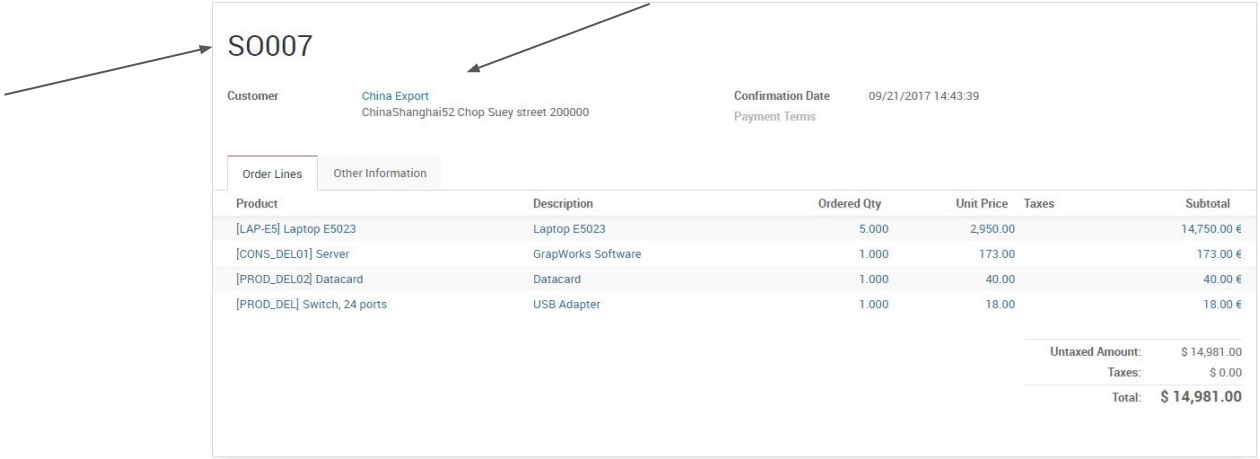
What is a database ?

Comparison with the Excel file

- Sheets are models
- Columns are fields
- Rows are records

id	name	city	email
base.res_partner_1	ASUSTeK	Taipei	asusteK@yourcompany.example.com
base.res_partner_address_27	Arthur Gomez	Taipei	
base.res_partner_address_12	James Miller	Taipei	
base.res_partner_address_2	Joseph Walters	Taipei	joseph.walters@asustek.com

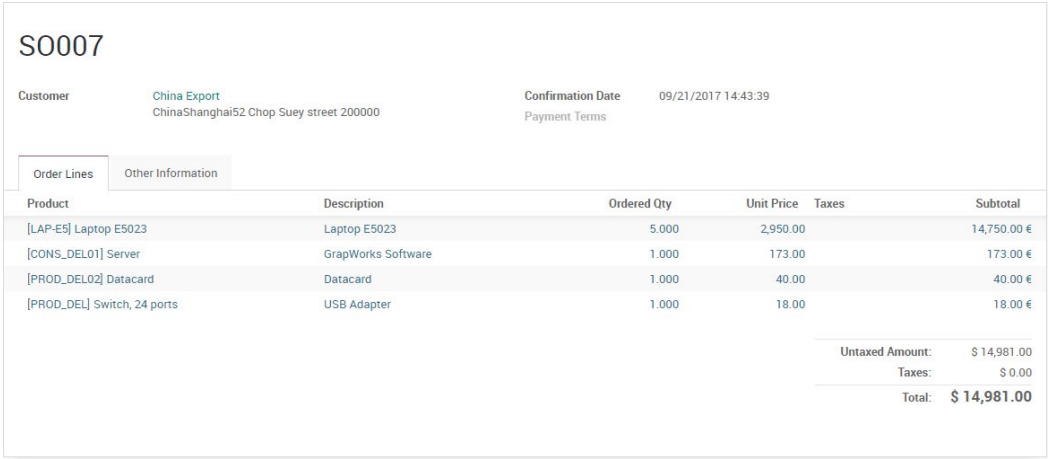
Models



Each Odoo model is a distinct translation of a business concept.
E.g. : An invoice, a sale order line, a customer.

But they can work together and have relationships through their fields

Fields


SO007

Customer: China Export
ChinaShanghai52 Chop Suey street 200000

Confirmation Date: 09/21/2017 14:43:39
Payment Terms:

Order Lines | Other Information

Product	Description	Ordered Qty	Unit Price	Taxes	Subtotal
[LAP-E5] Laptop E5023	Laptop E5023	5.000	2,950.00		14,750.00 €
[CONS_DEL01] Server	GrapWorks Software	1.000	173.00		173.00 €
[PROD_DEL02] Datacard	Datacard	1.000	40.00		40.00 €
[PROD_DEL] Switch, 24 ports	USB Adapter	1.000	18.00		18.00 €

Untaxed Amount: $ 14,981.00
Taxes: $ 0.00
Total: $ 14,981.00

Fields are where the data is stored
Eg : Name, Status, Price, Weight, …

They have different types & features (e.g. : Date, Dropdown menu, …)

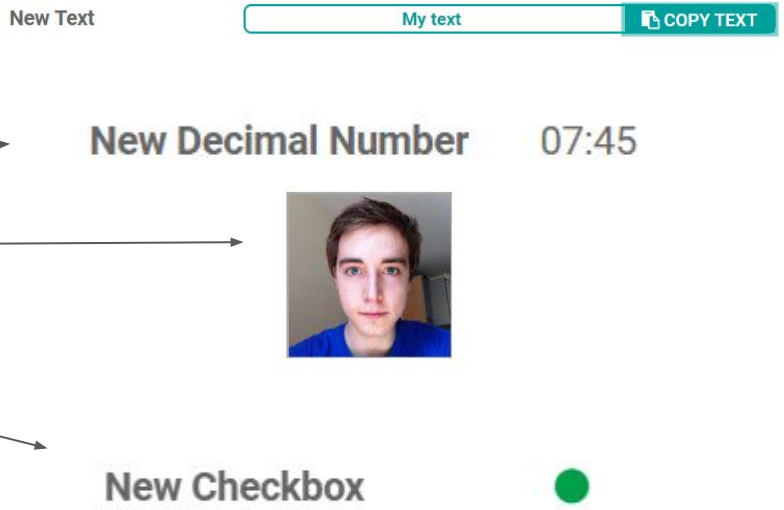
NB : Do not forget the naming conventions (“_id” & “_ids”)

Fields types

Text (char)	Selection (selection)
Multi-line Text (text)	HTML (html)
Integer number (integer)	File (binary)
Decimal number (float)	Many2One (many2one)
Date (date)	One2many (one2many)
Date & time (datetime)	Many2Many (many2many)
Checkbox (boolean)	

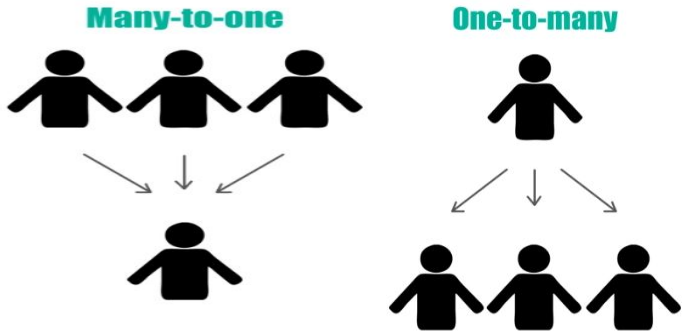
Fields (types) + Widgets

Text (char) + Copyclipboardchar
Decimal number (float) + timefloat
Binary + image
Checkbox (boolean) + toogle_button



Relational fields

Many2One & One2Many

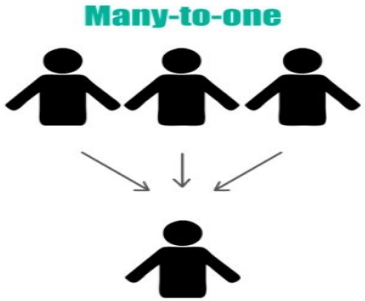


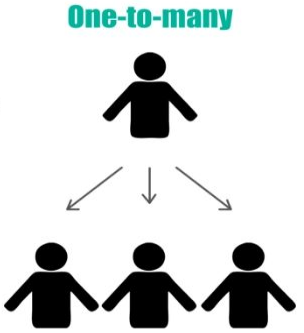
- M2O -> Many records pointing towards another one.

- O2M -> Does not exist “per se”, reverse search (M2O)

Relational fields

Many2Many & One2One



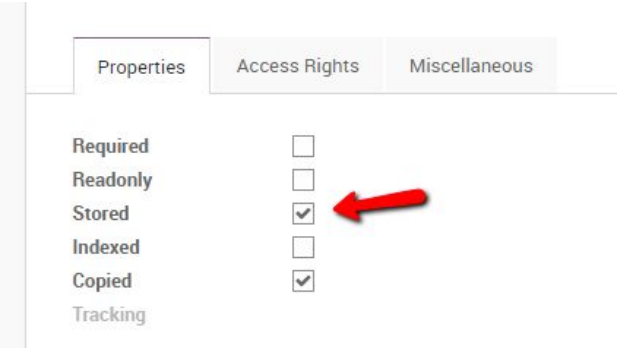


- M2M : Many records pointing towards many records… And vice versa !
- O2O : Does not exist in Odoo, use M2O instead

Stored and not stored

- *Is this information recorded in the DB or not?*
  - → YES: Stored
  - → NO: Not stored

- **All newly created fields are stored except the computed / related fields**
  - → To store them



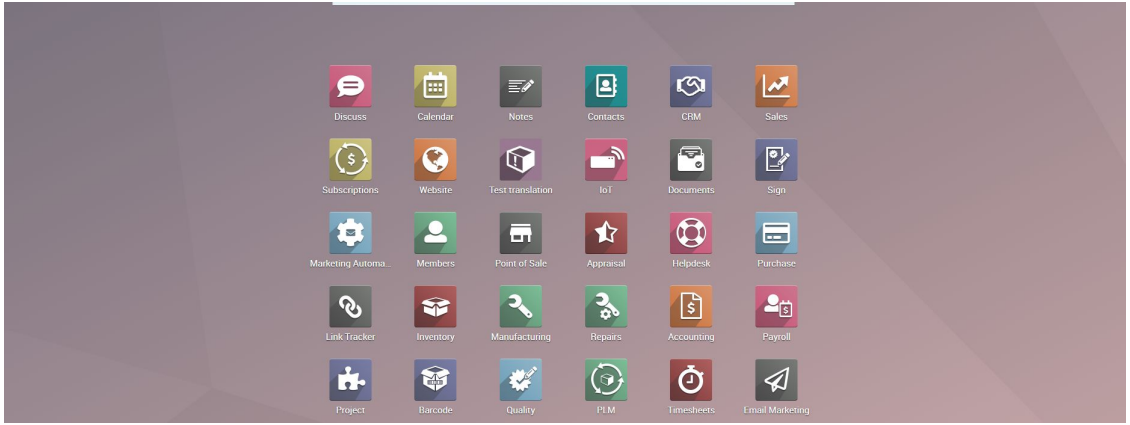
Stored and not stored

- ***What if the field is not stored?***
  - → It's not searchable
  - → No *Group by* possible (not available in the pivot view)
  - → Not usable in record rules (to be developed during the *Access Rights* session)
- **Why should I store an information?**
  - → Performance

⇒ If you have any doubt, you can always ask to a technically skilled colleague

Menu items

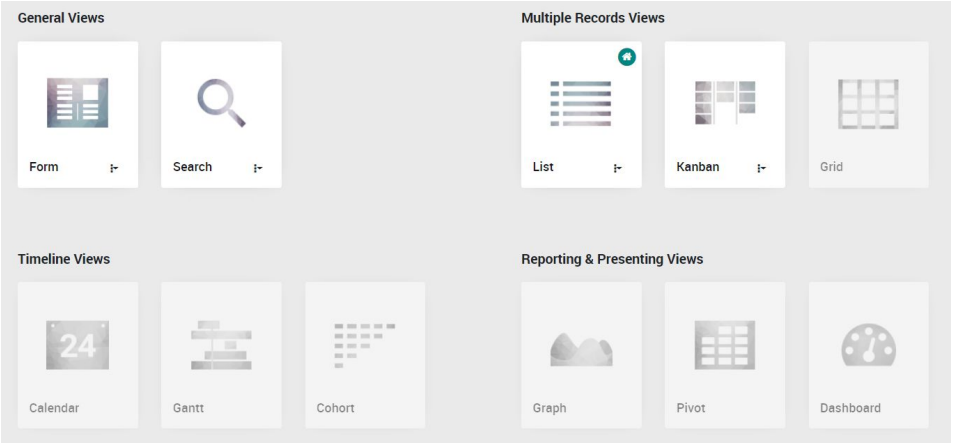
- Menus allow to navigate through the differents models



3

Technical Concepts : Studio & Views

Views



- Views are the interface that allows us to see the data in different ways (a graph, a form, a calendar, …)

- Different view types exist and can be grouped in ‘families’
- The language used to create those views is (mostly) XML
- They’re stored in the database (Debug -> Technical -> Views)

Basics of XML

Entity

- **Some characters have special meaning in XML**
- ***If you need to use them outside their initial use in the XML syntax, it will generate an error.***

- ***To avoid these errors, replace the initial character with an entity reference***

```
<message>salary < 1000</message>  ⟶  <message>salary &lt; 1000</message>
```

Entity reference	Character	Meaning
<	<	Less than
>	>	Greater than
&	&	Ampersand
'	‘	Apostrophe
"	“	Quotation mark

Basics of XML

XML & Markup

- **What does XML means?**
- ***eXtensible Markup Language.***

  → It's a markup language just like HTML.

- **Markup structure**
  - ***All XML elements must have a closing tag.***

```
<p>This is a paragraph.</p>
<br />
```

  - ***XML tags are case sensitive***

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

  - ***XML elements must be properly nested***

```
<b><i>This text is bold and italic</i></b>
```

  - ***XML can be commented***

```
<!-- This is a comment -->
```

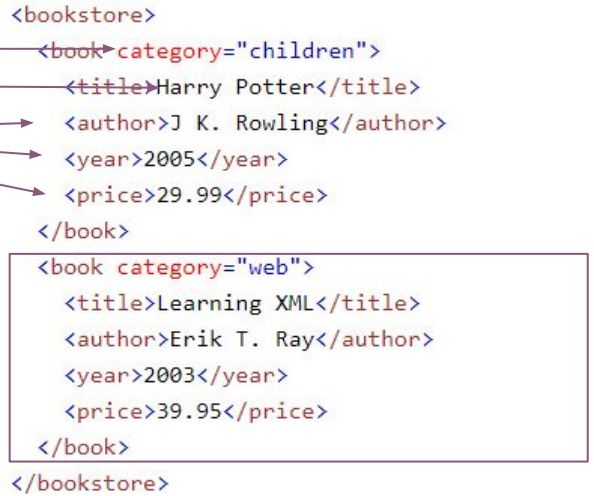
Basics of XML

Element

- **What is an XML element?**

- ***It's everything from the element's start tag to the element's end tag.***

- ***An element can contain:***
  - → Attributes
  - → Text
  - → Other elements
  - → A mix of the above



Basics of XML

Attribute

- **What is an XML attribute?**

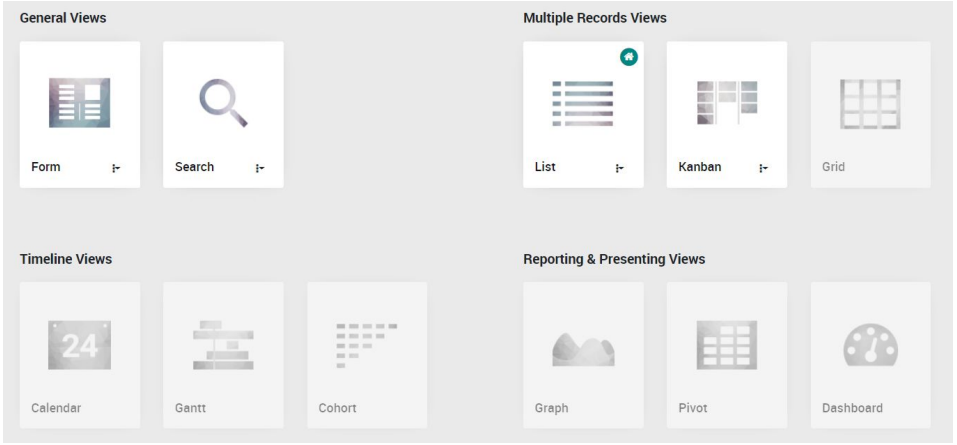
- *XML elements can have attributes that qualify them.*

- *XML attribute values must be quoted*

```
<note date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Views

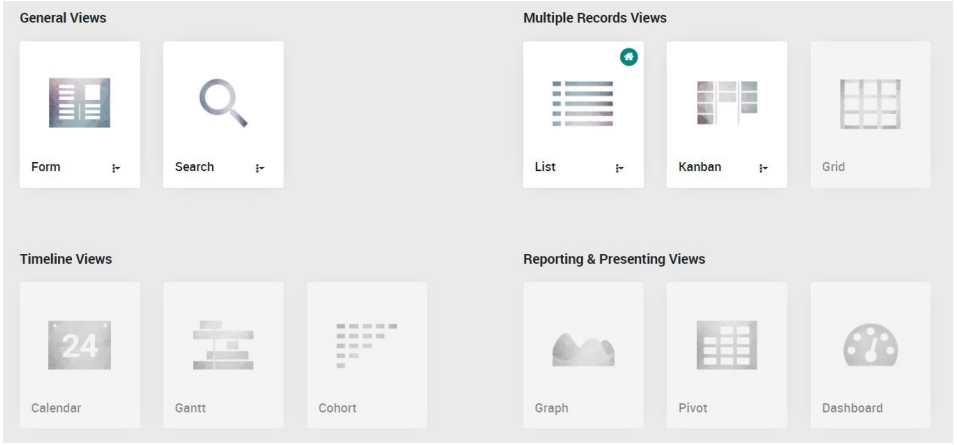
How should you choose ?



- Different usages
  - The Search view support all views and allow to filter/group/search
  - Gantt is for forecasting
  - Pivot, Graphs, … are for reporting purposes
  - Grid is either for mass creation or reporting

Views

How should you choose ?



- Different usages
  - Form views are used to create / edit complex records
  - List views to either provide overviews, look for records or edit simple records
  - Kanban view are built to support flows

General constraint :

A view is always based on a single model

- Contacts
- Opportunities
- Employees
- ...

Except... ?

Relational fields :

Example of a view showing several models


```
<form string="Sales Order" class="o_sale_order">
 ...
 <field name="partner_id">
 ...
 <page string="Order Lines" name="order_lines">
                        <field name="order_line" widget="section_and_note_one2many">
                          <tree string="Sales Order Lines" editable="bottom">
                              <field name="product_id"/>
                              <field name="qty_delivered"/>
                              <field name="price_total"/>
                          </tree>
```

View inheritance and Studio

- Standard views and updates
- Creating Inherited views (xpaths)
- Studio

View Inheritance

- **How an inherited view looks like?**
  - *A serie of change to be applied on the initial one.*
- **Main vs. Extension**
  - *A view can be modified by inherited view.*
  - *In our case, we'll always build the inheritance by creating "Extension" views based on a Main one*
- **Sequence**
  - *Odoo applies changes in a view respecting the sequence.*
    → The bigger the sequence is, the later your change occurs in the view
- **Change localization (WHERE)**
  - *XPATH vs. element*
  - *Keep it as simple as possible*
- **Positions (WHAT)**
  - *Before*
  - *After*
  - *Inside*

Attributes

How to make a view dynamic

- The same field cannot have different behaviors => Solution ?
- Clearer UI
- Enforce a process (! user frustration)

Attributes

A whole set of other tools

- Domains
- Default values
- Context
- Placeholders
- ...

Main Attributes

- **name**
- **readonly**
- **invisible**
- **groups**
- **attrs**
- **widget**
- **for (only for label)**
- **editable (only in tree view)**
- **domain**
- **placeholder**

Attributes : A few examples

- Sale orders
- Attachment
- Studio conditional attrs

```
<field name="validity_date" attrs="{'invisible': [('state', 'in', ['sale', 'done'])]}"/>
```

```
<field name="company_id" readonly="1"/>
```

```
<data>
  <xpath expr="//form[1]/sheet[1]/group[1]/group[2]/field[@name='title']" position="attributes">
    <attribute name="attrs">{"invisible": [["is_company","=",True]], "required": [["industry_id","=",1]]}</attribute>
  </xpath>
</data>
```

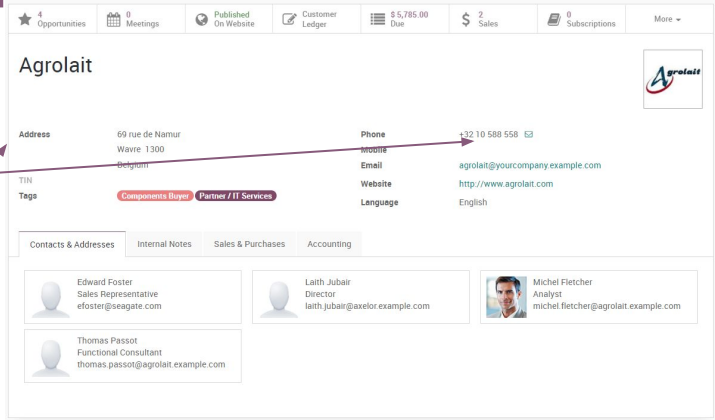
Odoo Studio Guided Tour

- Menus
- Views and options
- Fields and options (new and existing)

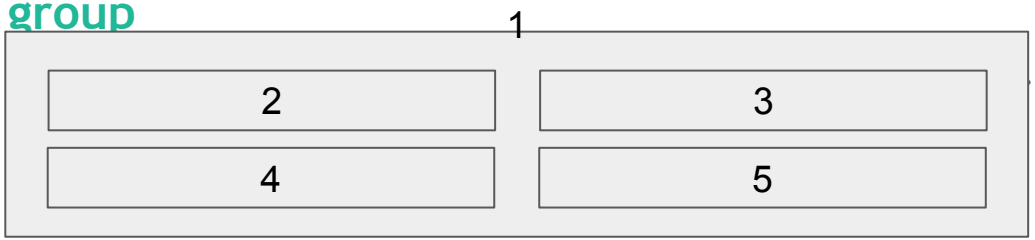
=> Explore

Main Elements in Odoo

- field
  - *Display a field and its label.*
- label
  - *Give a label to another element.*



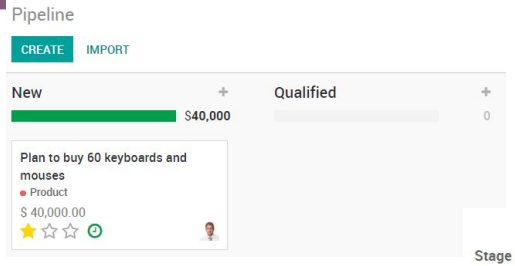
- group

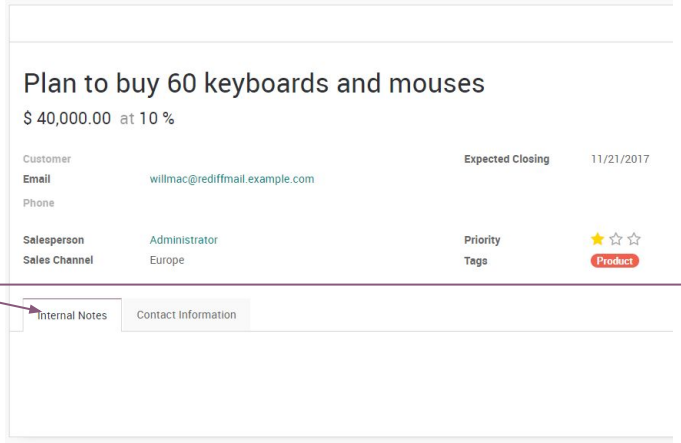



```
<group id="1">
    <group id="2">
    </group>
    <group id="3">
    </group>
    <group id="4">
    </group>
    <group id="5">
    </group>
</group>
```

Main Elements in Odoo

- **button**
  - *Display a button.*
- **separator**
  - *Give a name to a section.*
- **notebook**
  - *Display a sub-section in the form view.*
- **page**
  - *Display a tab in a notebook.*





Exercises

Industry 2

Business Need :

PLZ Pick-up the Phone© is a company active in the telemarketing industry. The main app they will work with is the CRM. For reporting as well as micro-management purposes, they would like to keep track of every call made by their employees.
The information they need are : WHO, WHEN, LENGTH OF THE CALL, STATUS, COMMENT
The different STATUS are : *Sold, Asked to call back, Fake number, Refused*
Of course, registering a new call should not erase the information from the previous one !

Functional Analysis :

On the opportunity :

- Create a new model “call report” and link it to opportunities
- Add a data table in a new tab which users can record their calls (date, length, status, comment) line by line
- Users should be able to edit/create their reports on those lines
- A menu item should allow the managers to access all “reports” and to group/filter them

Technical Analysis : Feasible with Studio

Industry 3

Business Need :

His&Her© is a fashion company that will heavily use the Mass Mailing module to promote their new catalogs. As such, they need to be sure of their customer gender (different campaigns). Moreover, they will heavily use titles in their email openings (ex : Miss, Mister, …) and would like to clearly distinguish those between genders (to avoid : Dear Miss John Smith).
For them, the titles are ordered in three categories (male, female, not defined).

Functional Analysis :

On the contacts

- Allow users to select a gender
- Make sure users can only select a title that matches the gender of the contact
- Both field should be mandatory as otherwise users won't fill them in
- Make sure users do not create new titles on the fly

Technical Analysis : Feasible with Studio

Don’t forget, if you ever have a question...

odoo 2020

PROFESSIONAL SERVICES

Thank you.

#TfFC