

Odoo Javascript Framework

The state of the web client

Géry Debongnie • Lead Developer • RD Framework Team

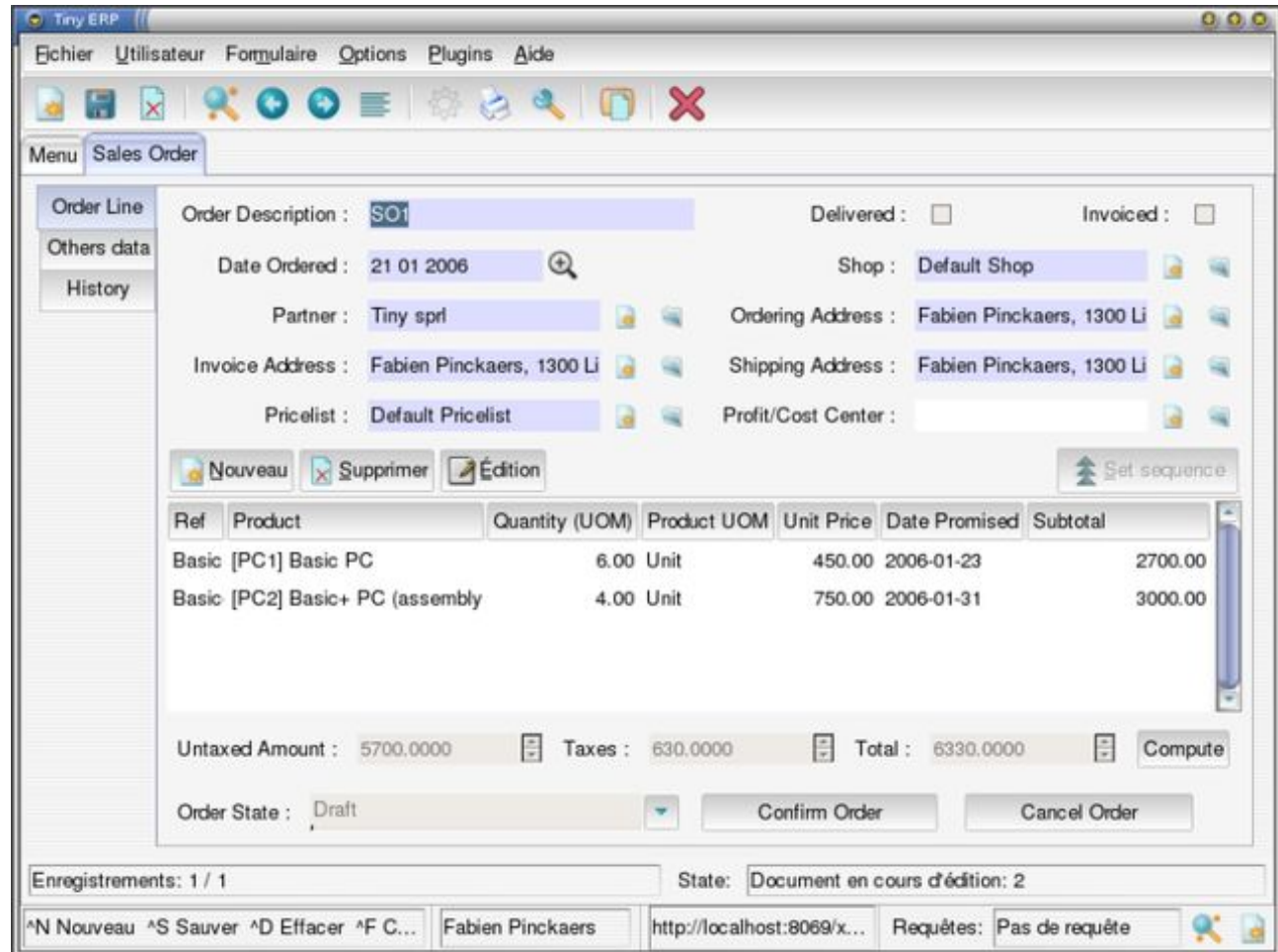
- 1 The Past: a brief history of Odoo web client
- 2 The Present: strengths and weaknesses
- 3 The Future: new ideas and design principles



1

The Past: A Brief history of Odoo client

<1 million years: GTK app with python server



<1 million years: GTK app with python server

500 000 years ago: GTK client + actual web client

The screenshot displays the OpenERP web client interface. At the top, the header includes the OpenERP logo, the company name 'Your Company (IntelCTSeed_demo)', the user role 'Administrator', and navigation icons for home, settings, and logout. Below the header is a red navigation bar with buttons for 'SALES', 'WAREHOUSE', 'ACCOUNTING', and 'SETTINGS'. A left sidebar contains a menu with options like 'Sales', 'Address Book', 'Products', 'Reporting', and 'Configuration'. The main content area is titled 'Products' and shows a form for editing a product named 'Ice Cream'. The form includes fields for 'Name', 'Codes' (with a 'Reference' dropdown set to '1'), and 'Characteristics' (with checkboxes for 'Can be Sold' and 'Can be Purchased'). Below these are tabs for 'Information', 'Procurement & Locations', 'Suppliers', 'Descriptions', and 'Accounting'. The 'Information' tab is active, showing 'Procurement' details (Product Type: Stockable Product, Supply method: Buy) and 'Base Prices' (Sale Price: 100.00, Cost Price: 70.00). It also shows 'Stocks' (Quantity On Hand: 0.000, Quantity Available: 0.000) and 'Unit of Measure' (Default Unit Of Measure: kg). A right sidebar contains sections for 'Attachments', 'Reports', 'Actions', 'Links', and 'Other Options'. A large purple arrow on the left points downwards. A red line points from the text 'Form View' to the main form area. A blue line points from the text 'By Default view is Simplified where only essential fields are displayed.' to the form area.

Form View

By Default view is Simplified where only essential fields are displayed.



<1 million years: GTK app with python server

500 000 years ago: GTK client + actual web client

100 000 years ago: GTK app is dropped, web client evolves






<1 million years: GTK app with python server

500 000 years ago: GTK client + actual web client

100 000 years ago: GTK app is dropped, web client evolves

2013--2014: version 8: assets bundle, CMS, POS



```
<template id="web.assets_common">
  <link rel="stylesheet" type="text/css" href="/web/static/lib/jquery.ui/jquery-ui.css"/>
  <link rel="stylesheet" type="text/less" href="/web/static/src/less/fonts.less"/>
  <link rel="stylesheet" type="text/less" href="/web/static/src/less/navbar.less"/>

  <script type="text/javascript" src="/web/static/lib/underscore/underscore.js"></script>
  <script type="text/javascript" src="/web/static/lib/jquery/jquery.js"></script>
  <script src="/web/static/lib/bootstrap/js/dropdown.js"></script>
</template>
```



<1 million years: GTK app with python server

500 000 years ago: GTK client + actual web client

100 000 years ago: GTK app is dropped, web client evolves

2013--2014: assets bundle, CMS, POS (v8)

early 2015: Odoo JS modules

```
odoo.define('web.ajax', function (require) {  
  "use strict";  
  
  var core = require('web.core');  
  var utils = require('web.utils');  
  var time = require('web.time');  
  
  ...  
  
  return {...};  
  
});
```




<1 million years: GTK app with python server

500 000 years ago: GTK client + actual web client

100 000 years ago: GTK app is dropped, web client evolves

2013--2014: assets bundle, CMS, POS (v8)

early 2015: Odoo JS modules

Mid-late 2015: v9

- New design
- Split community/enterprise
- Move toward business apps
- Large refactorings





<1 million years: GTK app with python server

500 000 years ago: GTK client + actual web client

100 000 years ago: GTK app is dropped, web client evolves

2013--2014: assets bundle, CMS, POS (v8)

early 2015: Odoo JS modules

Mid-late 2015: new design, split community/enterprise, ... (v9)

2016: v10

- Stabilisation,
- performance improvements,
- Onboarding,
- Odoo Studio



Customer Invoices / INV/2016/0003

EDIT CREATE

Print Action

1 / 1 < >

SEND BY EMAIL PRINT REGISTER PAYMENT REFUND INVOICE

DRAFT OPEN PAID

INV/2016/0003

Customer [Agrolait](#)
69 rue de Namur
1300 Wavre
Belgium

Invoice Date 10/08/2016
Salesperson [Administrator](#)

Payment Terms [30 Net Days](#)

INVOICE LINES OTHER INFO

Product	Description	Account	Quantity	Unit Price	Taxes	Amount
[CONS_DELO1] Server	[CONS_DELO1] Server raid 10 2048ECC ram	200000 Product Sales	5.000	90.00		\$ 450.00
[CONS_DELO3] Basic Computer	[CONS_DELO3] Basic Computer Dvorak keyboard left-handed mouse	200000 Product Sales	5.000	15.00		\$ 75.00

Untaxed Amount :	\$525.00
Tax :	\$0.00
Total :	\$525.00
<hr/>	
Amount Due :	\$525.00

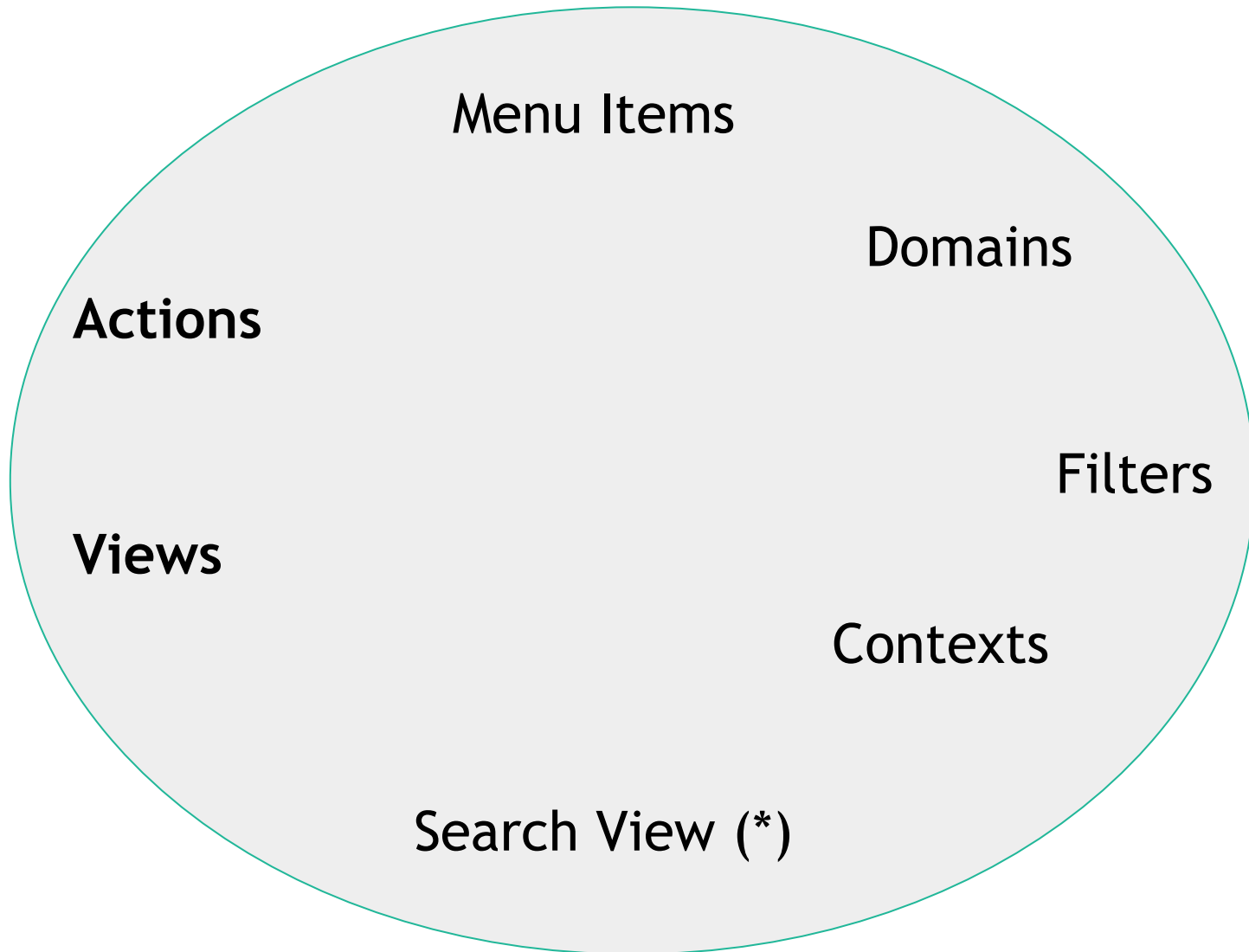


2

The Present: Strengths and weaknesses



Strength: good abstractions



Strength: extreme modularity

(almost) anything can be customized in a module

- Form view
- Menus
- Business logic
- App switcher background
- Logging
- import/export
- ...



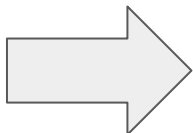
Example: the new web client (enterprise) is implemented in `addon web_enterprise/`



Strength: Code/Interface is data

Code/Interface = data

views	Search views
actions	Menu items
(custom) models	(custom) fields
Security rules	templates
reports	...



Can be modified dynamically



With great power comes great
responsibility

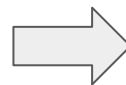
– Spider Man (or maybe Voltaire?)

Weakness: extreme modularity

Odoo modules = monkey patching

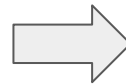


Any module can change anything



Hard to have a coherent system

Exponential number of combinations



Most of them untested

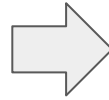


Other challenges

Dead code	Hard to test	Difficult to optimize
Api always changing	Mostly undocumented	Mostly untested
Coupling between view and model	Mobile experience not as good as it should be	Cannot instantiate formview in website
Some customs are hard (control panel)	Fragile => lots of combinations	flickering
Dead css	Cannot reuse form widgets	...

Why Odoo Studio is a big deal (for the web client)

Need to separate rendering from the rest



Good excuse to start refactoring everything!



Work on rewriting the (JS) views:

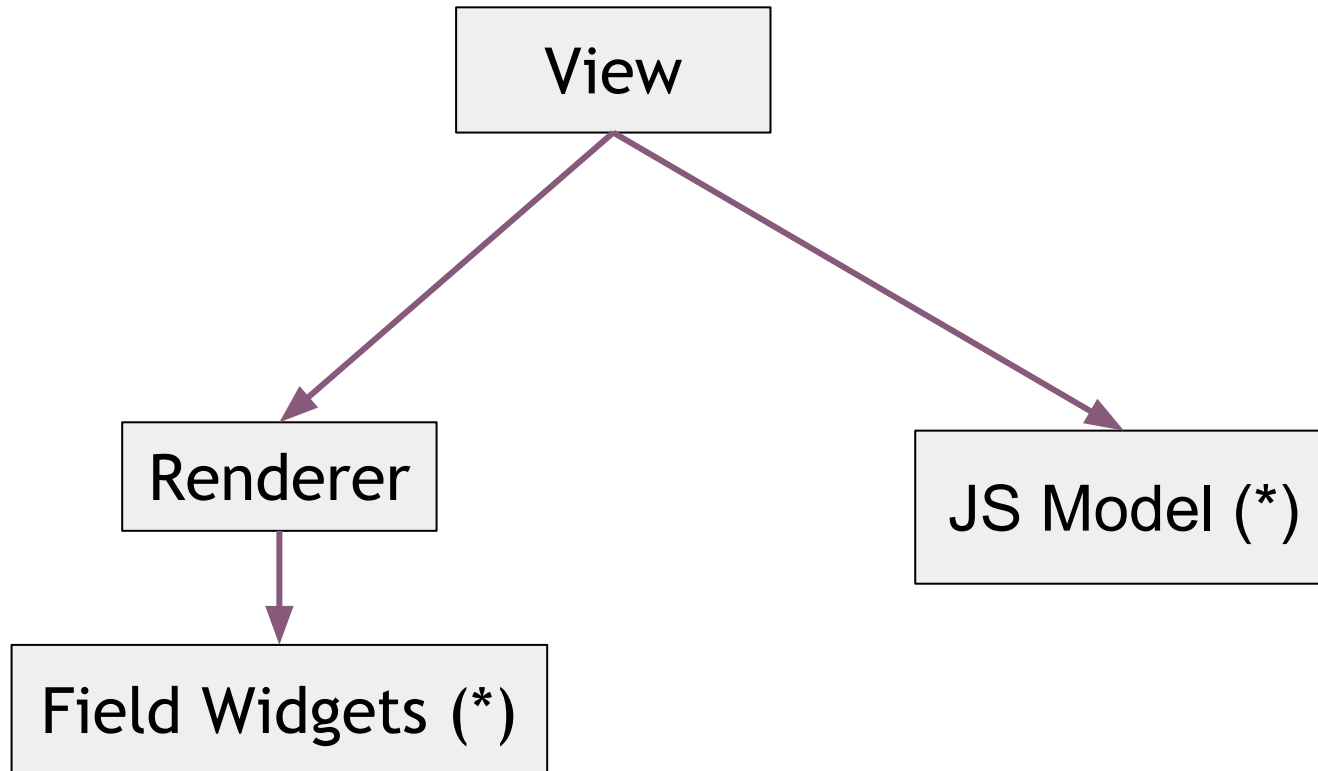
- Separate rendering from rest
- Solving some of those previous challenges
- Unify the views (widget and data fetching)



The Future: New Ideas and design principles



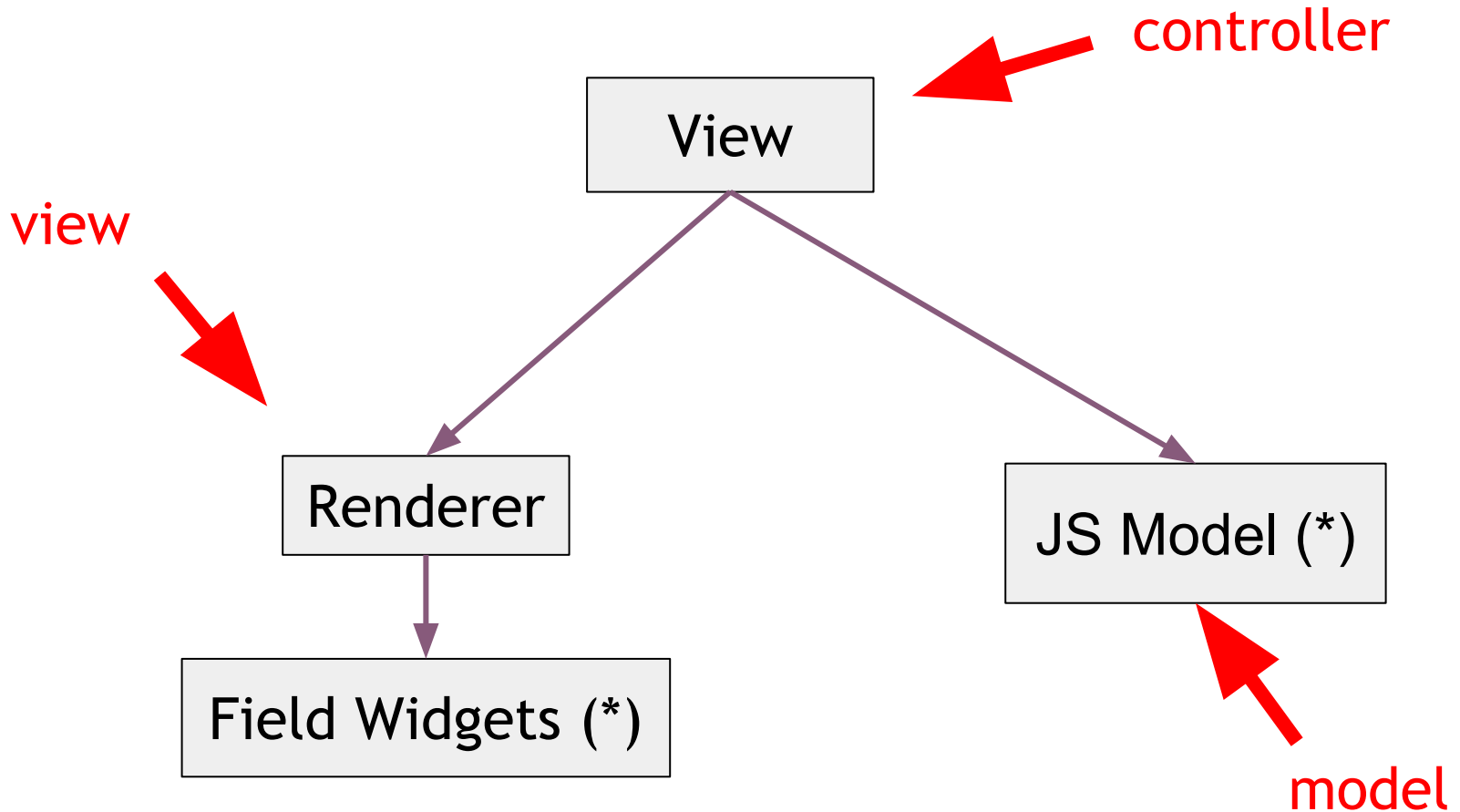
New views



(*) Shared between views



New views



(*) Shared between views

New views

Benefits: unit testable, no more flickering, about 40% less LOC

```
define_case('form view can switch to edit mode', function(assert) {  
  var form = render_view({  
    View: FormView,  
    arch: '<form string="Partners">' +  
      '<sheet>' +  
        '<group>' +  
          '<field name="f4"/>' +  
        '</group>' +  
      '</sheet>' +  
      '</form>',  
    res_id: 14,  
  });  
  
  assert.equal(form.mode, 'readonly', 'form view should be in readonly mode');  
  form.$buttons.find('.o_form_button_edit').click();  
  assert.equal(form.mode, 'edit', 'form view should be in edit mode');  
});
```



Design principles

- Move state away from DOM (and into JS)
- Separate rendering from fetching/processing data
- Rendering should be synchronous
- Delegate side effects by triggering events up the component tree



Solved challenges (maybe)

Dead code	Hard to test	Difficult to optimize
Api always changing	Mostly undocumented	Mostly untested
Coupling between view and model	Mobile experience not as good as it should be	Cannot instantiate formview in website
Some customs are hard (control panel)	Fragile => lots of combinations	flickering
Dead css	Cannot reuse form widgets	...



Roadmap (unofficial)

- Complete rewrite of views: form, list, kanban, ...
- Refactor rest of the web client (view manager/action manager / web client / session)
- Rethink widget life cycle
- Unit test as much as possible
- Document core classes

Some other ideas

- Use Ecmascript 2015
- Use some virtual dom library
- Rewrite server in nodejs (*)
- Use http/2 multiplexing and handle static assets better
- ...

Thank you.



#odooexperience