

# Make Your App Frontend Compatible

*From trakabacks to Hello World*

Thibault DELAVALLEE  • R&D Engineer

<https://www.odoo.com/documentation>



[https://github.com/tde-banana-odoo/odoo\\_plants](https://github.com/tde-banana-odoo/odoo_plants)



Going on the World  
Wide Internet Y2K

# Hello World (Wide Web)

☰ Plant Nursery

PlantsCustomersOrdersSettings

🔍 Administrator (\$1)

Plants


Search...

CREATEIMPORT

FiltersGroup ByFavorites1-9 / 9

Apple Tree


Fruit Tree



AlcoholFruitPromo

Beaucarnea Recurvata


Palm



Evergreen

Cherry Tree


Fruit Tree



AlcoholFruit

Dracaena Marginata


Palm



Evergreen

Lemon Potted Tree


Potted Tree



AlcoholFruitPromo

Orange Potted Tree


Potted Tree



AlcoholFruit

Pepper Potter Tree


Potted Tree



AlcoholFruit

Plum Tree


Fruit Tree



AlcoholFruit

Ravena Rivularis

Palm



EvergreenFruit

Hello World



Routing



Templates

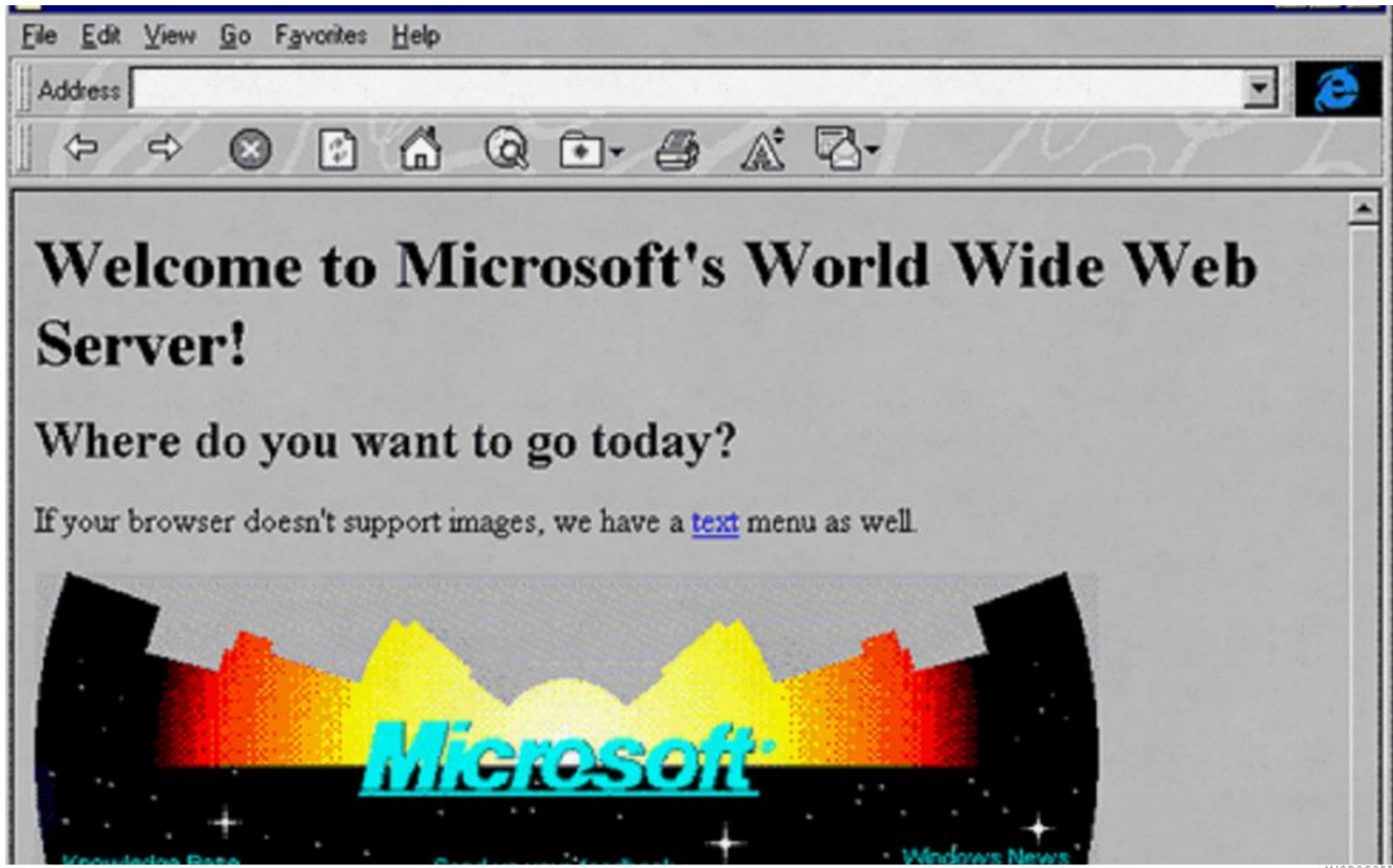


JSON



Next

# Hello World (Wide Web)



1 Hello World !

2 **Routing**

3 Templates

4 JSON

5 What's next ?



2

## Routing: From URLs to methods

# Routing

← → ↻ ⓘ localhost:8069/plants?category=evergreen



odoo



/plants?category=evergreen

```
@route('/plants')
def plants_index(self, **post):
    # compute stuff
    Category = post.get('category')
    return Response()
```

```
post = dict({
    'category': 'evergreen',
    ...
})
```





# Routing: Controller

- Controller for Plant Nursery: `odoo.http.Controller` class
  - routes definitions
  - `tools` method, prepare data

```
from odoo.http import request, route, Controller

class PlantController(Controller):

    def _check_access(self, stuff):
        return has_access

    def _prepare_values(self, post):
        values = dict(post)
        post.update({})
        return values

    @route('/plants')
    def plants_index(self, **post):
        values = self._prepare_values(**post)
        return request.render("index", values)
```



# Routing: Controller

- Controller for Plant Nursery: `odoo.http.Controller` class
  - routes definitions, tools method, prepare data
  - business code should always be in models

```
from odoo.http import request, route, Controller

class PlantController(Controller):

    def update_plant_order(self, plants):
        for plant in plants:
            for order in plant.orders.filtered(lambda o: o.state == 'open'):
                order.write(stuff)
        return True

    @route('/plants/update')
    def plants_update(self, **post):
        plants = ...
        self.update_plant_order(plants)
        return request.render("index", plants)
```



# Routing: Controller

- Controller for Plant Nursery: `odoo.http.Controller` class
  - routes definitions, tools method, prepare data
  - business code should always be in models

```
from odoo.http import request, route, Controller

class PlantController(Controller):

    def update_plant_order(self, plants):
        for plant in plants:
            for order in plant.orders.filtered(lambda o: o.state == 'open'):
                order.write({'state': 'open'})
            return True

    @route('/plants/update')
    def plants_update(self, **post):
        plants = ...
        plants.update_order()
        return request.render("index", plants)
```



# Routing: route

- Controller for Plant Nursery: `odoo.http.Controller` class
- Route definitions to map URLs to controller methods
  - `@route('/plants')`  $\leftrightarrow$  `mydomain/plants`

```
@route('/plants')
def plants_index(self):
    values = {
    }
    return request.render("index", values)
```



# Routing: route

- Controller for Plant Nursery
- Route definitions to map URLs to controller methods
  - `@route('/plants')` -> `mydomain/plants`
  - HTTP route: mapping and parameters

```
@route(['/plants', '/plants/page/<int:page>'],
      type='http')
def plants_index(self, page=1, category=None):
    values = {
        'page': page,
        'category': category,
    }
    return request.render("index", values)
```



# Routing: route

- Controller for Plant Nursery
- Route definitions to map URLs to controller methods
  - `@route('/plants')` -> `mydomain/plants`
  - HTTP route: mapping and parameters
  - authentication: user, public, none ! **access rights**

```
@route(['/plants', '/plants/page/<int:page>'],
      type='http',
      auth='public')
def plants_index(self, page=1, category=None):
    values = {
        'page': page,
        'category': category,
    }
    return request.render("index", values)
```



# Routing: route

- Controller for Plant Nursery
- Route definitions to map URLs to controller methods
  - `@route('/plants')` -> `mydomain/plants`
  - HTTP route: mapping and parameters
  - authentication: user, public, none
  - post kwargs

```
@route(['/plants', '/plants/page/<int:page>'],
      type='http',
      auth='public')
def plants_index(self, page=1, category=None, **post):
    values = {
        'page': page,
        'category': category,
        'available': post.get('available', True),
    }
    return request.render("index", values)
```



# Routing: request

- Controller for Plant Nursery
- Route definitions to map URLs to controller methods
- `odoo.http.request`
  - `env, session`: compute / fetch data
  - `render`: rendered template as Response

```
def plants_index(self, page=1, category=None, **post):
    if request.session.uid: # logged
        plant_domain = []
    else:
        plant_domain = [('published', '=', True)]
    if category:
        plant_domain += [('category', '=', category)]
    plants = request.env['plant.plant'].search(plant_domain)
    values = {
        'page': page,
        'category': category,
        'plants': plants,
    }
    return request.render("index", values)
```





1 Hello World !

2 Routing

3 **Templates**

4 JSON

5 What's next ?



Templates: QWeb for  
dynamic content

# Templates

- QWeb template: request.render() based on values

```
def plants_index(self, page=1, category=None, **post):  
    values = {'name': 'world'}  
    return request.render('index', values)
```

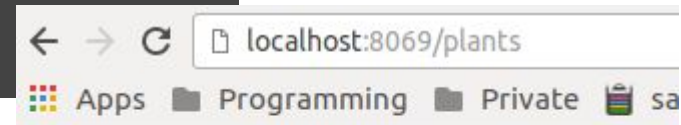
```
<template id="plant_hello" name="Hello World">  
    <html>  
        <body>  
            <h1>Hello <t t-esc="name"/>!</h1>  
        </body>  
    </html>  
</template>
```



# Templates

- QWeb template: request.render() based on values

```
def plants_index(self, page=1, category=None, **post):  
    values = {'name': 'world'}  
    return request.render('index', values)
```



**Hello world!**

```
<template id="plant_hello" name="Hello World">  
  <html>  
    <body>  
      <h1>Hello <t t-esc="name"/>!</h1>  
    </body>  
  </html>  
</template>
```



# Templates: assets

- QWeb template: request.render()
  - add your assets to Odoo assets (backend, frontend, common)
  - improve templates (**inherit**) / reuse templates (**t-call**)

```
<template id="plant_hello" name="Hello World">
  <t t-call="web.layout">
    <div id="wrapwrap" class="row">
      <h1 class="col-md-12">Hello <t t-esc="name"/>!</h1>
    </div>
  </t>
</template>

<template id="assets_frontend" inherit_id="web.assets_frontend"
  name="My Assets Addition">
  <xpath expr="//link[last()]" position="after">
    <link rel="stylesheet" type="text/less"
      href="/plant_nursery/static/src/less/frontend.less"/>
    <link rel="text/javascript" type="text/less"
      href="/plant_nursery/static/src/less/frontend.js"/>
  </xpath>
</template>
```



Bootstrap

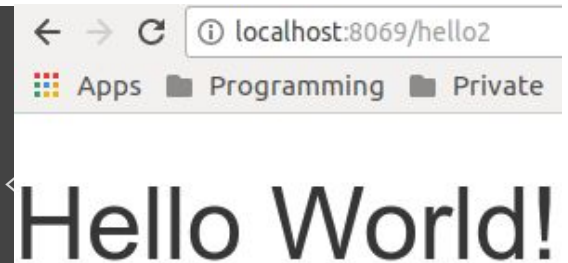


# Templates: assets

- QWeb template: request.render()
  - add your assets to Odoo assets (backend, frontend, common)
  - improve templates (**inherit**) / reuse templates (**t-call**)

```
<template id="plant_hello" name="Hello World">
  <t t-call="web.layout">
    <div id="wrapwrap" class="row">
      <h1 class="col-md-12">Hello <t t-esc="name"/>!</h1>
    </div>
  </t>
</template>

<template id="assets_frontend" inherit_id="web.assets_frontend"
  name="My Assets Addition">
  <xpath expr="//link[last()]" position="after">
    <link rel="stylesheet" type="text/less"
      href="/plant_nursery/static/src/less/frontend.less"/>
    <link rel="text/javascript" type="text/less"
      href="/plant_nursery/static/src/less/frontend.js"/>
  </xpath>
</template>
```



# Templates: QWeb

- QWeb template: `request.render()`
  - variables: **t-esc**, **t-set / t-value**, Odoo records: **t-field**
  - attributes: **t-att-<attr>=#{}**
  - control flow: **t-if**, **t-else**, **t-foreach / t-as**

```
<t t-call="web.layout">
  <div id="wrapwrap" class="row">
    <h1>Hello <t t-esc="name"/>!</h1>
    <div class="col-md-10">
      <p>Welcome to <strong>
        <span t-field="company.name"/>'s Plant Store</strong>
      </p></div>
      <ul t-foreach="plants" t-as="plant">
        <li class="col-md-3">
          <a t-attf-href="/plant/#{plant.id}">
            <span t-field="plant.name"/>
          </a>
        </li>
      </ul>
    </div>
  </t>
```



# Templates: QWeb

- QWeb template: `request.render()`
- QWeb = XML compliant templating language



## Hello world!

Welcome to **Woody Cutters's Plant Store**

- Apple Tree
- Lemon Potted Tree
- Ravena Rivularis
- Beaucarnea Recurvata
- Orange Potted Tree





# Templates



Welcome to **Woody Cutters's** Plant Store !



## Select a Plant



Apple Tree



Beaucarnea  
Recurvata



Cherry Tree

## Ask for Quote

Send an email to [our store](#)

Hello World



Routing



Templates



JSON



Next

1 Hello World !

2 Routing

3 Templates

4 **JSON Routes**

5 What's next ?

# 4

## Intro to JSON routes and Widgets

# JSON: routes

- JSON routes
  - asynchronous communication
  - returns JSON-ified data

```
@route('/plant/get_data', type='json', auth='public')
def plant_get_data(self):
    plants = request.env['plant.plant'].search([])
    return {
        'promo_plants': plants.filtered(...),
        'free_plants': plants.filtered(...),
    }
```



# JSON: Widget

- JSON routes: call to fetch / process data asynchronously
- JS Widget: Odoo Web Widget
  - define templates, events, lifecycle methods

```
var Widget = require("web.Widget");

var QuoteAsk = Widget.extend({
  template: 'plant_nursery.quote_ask',
  events: {
    'xmas_button': 'resetQuote',
  },

  start: function () {
  },
});
```



# JSON: RPC (aka async nightmare)

- JSON routes: call to fetch / process data asynchronously
  - JS Widget: Odoo Web Widget
    - rpc.query to call server
- asynchronous -> Deferred !

```
var rpc = require('web.rpc');
var qweb = require('web.core.qweb');

fetchData: function () {
    return rpc.query({
        route: '/plant/get_data',
        params: {
        },
    }).then(function (data) {
        var $plants = qweb.render("plant.plants", {widget: data});
        self.$('.o_content').replaceWith($plants);
    });
}
```



# JSON: RPC (aka async nightmare)

- JSON routes: call to fetch / process data asynchronously
- JS Widget
  - rpc.query to call server
  - QWeb also used for rendering

```
var rpc = require('web.rpc');
var qweb = require('web.core.qweb');

fetchData: function () {
  return rpc.query({
    route: '/plant/get_data',
    params: {
    },
  }).then(function (data) {
    var $plants = qweb.render("plant.plants", {widget: data});
    self.$('.o_content').replaceWith($plants);
  });
}
```



# JSON

- JSON routes
- JS Widget -> **see Web framework and presentations**

## XMas Surprise Quote

Your Name

Your Email

Promo

☐ Lemon Potted Tree **50 % OFF**

Free

☐ Pepper Potter Tree **100 % OFF**

I want another

Get XMas





- 1 Hello World !
- 2 Routing
- 3 Templates
- 4 JSON
- 5 What's next ?



5

Going further: Website  
and CMS by Martin

5

HeY n WeLcOmE 2 *timehop*

# Abe's Page

since 1995

cOOl LiNkS

[Timehop Web Page](#) 

[aBoUt AbE](#) 

[Geocities \(MaKe Ur OwN WEBSIGHT!!\)](#)

[SIGN my GuEsT bOOk](#) 

[Gallery](#)



UpDaTeD SePt. 12, 1999 !!

 u are visitor number **000000034** 

 email the WeBmAsTeR  
[cute\\_abe\\_95@hotmail.com](mailto:cute_abe_95@hotmail.com) 

# Thank you !



[https://github.com/tde-banana-odoo/odoo\\_plants](https://github.com/tde-banana-odoo/odoo_plants)



#odooexperience

Thibault DELAVALLEE 🍌 • R&D Engineer