



EBook Gratis

APRENDIZAJE

odoo-8

Free unaffiliated eBook created from
Stack Overflow contributors.

#odoo-8

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con odoo-8.....	2
Observaciones.....	2
Versiones.....	2
Examples.....	2
Preparar.....	2
Configuración:.....	2
¿Qué es Odoo?.....	7
Capítulo 2: ¿Cuáles son los métodos y detalles de ORM?.....	9
Observaciones.....	9
Examples.....	10
Diferentes tipos de métodos de ORM.....	10
Capítulo 3: Añadir archivos CSS y Javascript al módulo Odoo.....	11
Sintaxis.....	11
Parámetros.....	11
Observaciones.....	11
Examples.....	12
Almacena archivos CSS y JS correctamente en el módulo Odoo.....	12
Opción 1: [BACKEND] Agrega archivos CSS y Javascript para usar en páginas internas.....	12
Opción 2: [FRONTEND] Agrega archivos CSS y Javascript para usar en un sitio web público.....	12
Opción 3: [COMÚN] Agregue archivos CSS y Javascript para usar en todas las páginas (backen.....	13
Capítulo 4: Campos utilizados en Odoo 8.....	14
Introducción.....	14
Parámetros.....	14
Observaciones.....	14
Examples.....	16
Ejemplos de campos de Odoo 8.....	16
Capítulo 5: Cómo activar el modo de desarrollador OpenERP.....	17
Observaciones.....	17
Examples.....	18

Activar el modo desarrollador.....	18
Activando el modo desarrollador en Odoo 8.....	19
Activar el modo desarrollador en Odoo 10.....	19
Capítulo 6: Configurar correo electrónico - Office 365 en Odoo.....	21
Examples.....	21
Configurar E-Mail.....	21
Capítulo 7: Crear funciones automatizadas para el modelo.....	24
Introducción.....	24
Examples.....	24
En primer lugar, necesita crear un archivo xml para realizar una llamada a la función.....	24
Archivo Python correspondiente.....	24
Capítulo 8: RPC utilizando la API de Odoo v8 (función Call Python desde JavaScript).....	25
Observaciones.....	25
Examples.....	25
Un ejemplo de modelo de Odoo para llamar a métodos de.....	25
Ejemplos de Odoo RPC.....	26
Capítulo 9: Widgets personalizados para campos.....	28
Observaciones.....	28
Examples.....	28
Widget personalizado para campos numéricos para usar en TreeView.....	28
Creditos.....	30

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [odoo-8](#)

It is an unofficial and free odoo-8 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official odoo-8.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con odoo-8

Observaciones

Esta sección proporciona una descripción general de qué es odoo-8 y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de odoo-8, y vincular a los temas relacionados. Dado que la Documentación para odoo-8 es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Versiones

Número de versión	Comunidad	Empresa	Licencia	Fecha de lanzamiento
8.0	Sí	No	GNU AGPL	2014-09-18
9.0	Sí	Sí	GNU AGPL V3	2015-10-01

Examples

Preparar

Odoo se puede instalar de tres maneras diferentes:

1. Instaladores empaquetados (más fáciles, menos flexibles)
2. Fuente de instalación (tarda en instalarse, muy flexible)
3. Una imagen oficial de docker de [docker.com](https://www.docker.com)

Los paquetes oficiales con todos los requisitos de dependencia relevantes están disponibles en [odoo.com](https://www.odoo.com).

Windows

Descarga y ejecuta el [instalador](#).

Nota: en Windows 8 puede ver una advertencia titulada "Windows protege su PC". Haga clic en Más información y luego ejecútelo de todos modos. Acepte el indicador de UAC y siga los distintos pasos de instalación. Odoo se iniciará automáticamente al final de la instalación.

Configuración:

El archivo de configuración se puede encontrar en % PROGRAMFILES% \ Odoo 8.0-id \ server \ openerp-server.conf. (ID es su nombre de usuario del sistema)

El archivo de configuración se puede editar para conectarse a un Postgresql remoto, editar ubicaciones de archivos o configurar un dbfilter. Para volver a cargar el archivo de configuración, reinicie el servicio de Odoo a través de Servicios doodoo server.

Linux

Distribuciones basadas en Debian

Para instalar Odoo 8.0 en una distribución basada en Debian, ejecute los siguientes comandos como root:

```
# wget -O - https://nightly.odoo.com/odoo.key | apt-key add -  
# echo "deb http://nightly.odoo.com/8.0/nightly/deb/ ." >> /etc/apt/sources.list  
# apt-get update && apt-get install odoo
```

Esto instalará automáticamente todas las dependencias, instalará Odoo como demonio y lo iniciará automáticamente.

Tenga en cuenta que

para imprimir informes en PDF, debe instalar wkhtmltopdf usted mismo: la versión de wkhtmltopdf disponible en los repositorios de Debian no admite encabezados y pies de página, por lo que no puede instalarse automáticamente. La versión recomendada es 0.12.1 y está disponible en la página de descarga de wkhtmltopdf, en la sección de archivo. Como no hay una versión oficial para Debian Jessie, puede encontrar el paquete en <http://nightly.odoo.com/extra/> . o puede descargarlo e instalarlo desde la página de descarga de wkhtmltopdf como esta

```
# wget https://bitbucket.org/wkhtmltopdf/wkhtmltopdf/downloads/{path to correct distro and  
system architecture}  
# sudo dpkg -i {.deb package}  
# sudo cp /usr/local/bin/wkhtmlto* /usr/bin/
```

El archivo de configuración se puede encontrar en `/etc/odoo/openerp-server.conf`

Cuando se edita el archivo de configuración, Odoo debe reiniciarse usando el servicio:

```
$ sudo service odoo restart Reinicio de odoo: ok
```

Distribuciones basadas en RPM

Con las distribuciones basadas en RHEL (RHEL, CentOS, Scientific Linux), EPEL debe agregarse a los repositorios de la distribución para que todas las dependencias de Odoo estén disponibles. Para CentOS:

```
$ sudo yum install -y epel-release
```

Para otras distribuciones basadas en RHEL, consulte la documentación de EPEL.

A continuación se muestran los pasos de instalación.

```
$ sudo yum install -y postgresql-server
$ sudo postgresql-setup initdb
$ sudo systemctl enable postgresql
$ sudo systemctl start postgresql
$ sudo yum-config-manager --add-repo=https://nightly.odoo.com/8.0/nightly/rpm/odoo.repo
$ sudo yum install -y odoo
$ sudo systemctl enable odoo
$ sudo systemctl start odoo
```

Tenga en cuenta que

Para imprimir informes en PDF, debe instalar wkhtmltopdf usted mismo: la versión de wkhtmltopdf disponible en los repositorios de Fedora / CentOS no admite encabezados y pies de página, por lo que no puede instalarse automáticamente. Utilice la versión disponible en la página de descarga de wkhtmltopdf. Configuración, similar a debian se puede instalar con

```
wget https://bitbucket.org/wkhtmltopdf/wkhtmltopdf/downloads/{path to correct distro and
system architecture}
sudo rpm -i { .rpm package}
sudo cp /usr/local/bin/wkhtmlto* /usr/bin/
```

El archivo de configuración se puede encontrar en `/etc/odoo/openerp-server.conf`

Cuando se edita el archivo de configuración, Odoo debe reiniciarse a través de Systemd:

```
$ sudo systemctl restart odoo
```

Fuente de instalación

Odoo zip se puede descargar desde https://nightly.odoo.com/8.0/nightly/src/odoo_8.0.latest.zip , el archivo zip debe descomprimirse para usar su contenido.

Git permite una actualización más simple y un cambio más fácil entre diferentes versiones de Odoo. También simplifica el mantenimiento de parches y contribuciones que no sean de módulo. El principal inconveniente de git es que es significativamente más grande que un tarball, ya que contiene toda la historia del proyecto Odoo.

El repositorio de git es <https://github.com/odoo/odoo.git>.

Entonces puedes clonar el repositorio con

```
$ git clone https://github.com/odoo/odoo.git
```

Instalacion de dependencias

La instalación de origen requiere la instalación manual de dependencias:

Python 2.7. en Linux y OS X, incluido por defecto

en Windows, use el instalador oficial de Python 2.7.9.

Si Python ya está instalado, asegúrese de que sea 2.7.9, las versiones anteriores son menos convenientes y las versiones 3.x no son compatibles con Odoo

configurando PostgreSQL

Después de la instalación, deberá crear un usuario postgres: de forma predeterminada, el único usuario es postgres, y Odoo prohíbe la conexión como postgres.

en Linux, use el paquete de su distribución, luego cree un usuario postgres llamado como su nombre de usuario:

```
$ sudo su - postgres -c "createuser -s $USER"
```

Debido a que el inicio de sesión de rol es el mismo que el que usa para unix, los sockets de unix pueden usarse sin contraseña. en OS X, postgres.app es la forma más sencilla de comenzar, luego cree un usuario postgres como en Linux

en Windows, use PostgreSQL para Windows y luego agregue el directorio bin de PostgreSQL (predeterminado: C:\Archivos de programa\PostgreSQL\9.4\bin) a su PATH

cree un usuario postgres con una contraseña usando la interfaz de administración de pg: abra pgAdminIII, haga doble clic en el servidor para crear una conexión, seleccione Editar > Nuevo objeto R Nuevo rol de inicio de sesión, ingrese el nombre de usuario en el campo Nombre del rol (por ejemplo, odoo), luego abra la pestaña Definición e ingrese la contraseña (por ejemplo, odoo), luego haga clic en Aceptar.

El usuario y la contraseña se deben pasar a Odoo usando las opciones -w y -r o el archivo de configuración

Dependencias de Python enumeradas en el archivo Requirements.txt.

en Linux, las dependencias de Python pueden ser instalables con el administrador de paquetes del sistema o usando pip.

Para las bibliotecas que usan código nativo (Pillow, lxml, greenlet, gevent, psycopg2, ldap) puede ser necesario instalar herramientas de desarrollo y dependencias nativas antes de que pip pueda instalar las dependencias por sí mismo. Estos están disponibles en paquetes -dev o -devel para Python, Postgres, libxml2, libxslt, libevent, libsasl2 y libldap2. Luego se pueden instalar las dependencias de Python:

```
$ pip install -r requirements.txt
```

En OS X, deberá instalar las herramientas de línea de comandos (xcode-select --install) y luego descargar e instalar un administrador de paquetes de su elección (homebrew, macports) para instalar dependencias que no sean de Python. pip se puede usar para instalar las dependencias de Python como en Linux:

```
$ pip install -r requirements.txt
```

en Windows necesita instalar algunas de las dependencias manualmente, ajustar el archivo Requirements.txt y luego ejecutar PIP para instalar los remaning.

```
Install psycopg using the installer here http://www.stickpeople.com/projects/python/win-psycopg/
```

Luego edite el archivo Requirements.txt: elimine psycopg2 como ya lo tiene. elimine las opciones opcionales de python-ldap, gevent y psutil porque requieren compilación. agregue pypiwin32 porque es necesario en windows.

Luego use pip para instalar las dependencias usando el siguiente comando desde un indicador de cmd.exe (reemplace \YourOdoopath por la ruta real donde descargó Odoopath):

```
C:\> cd \YourOdoopath
C:\YourOdoopath> C:\Python27\Scripts\pip.exe install -r requirements.txt
```

Menos CSS a través de nodejs

en Linux, use el administrador de paquetes de su distribución para instalar nodejs y npm.

Tenga en cuenta que

En debian wheezy y Ubuntu 13.10 y antes de que necesite instalar nodejs manualmente:

```
$ wget -qO- https://deb.nodesource.com/setup | bash -
$ apt-get install -y nodejs
```

En versiones posteriores de Debian (> jessie) y ubuntu (> 14.04) es posible que deba agregar un enlace simbólico cuando los paquetes npm llaman a node, pero debian llama a los binarios nodejs

```
$ apt-get install -y npm
$ sudo ln -s /usr/bin/nodejs /usr/bin/node
```

Once npm is installed, use it to install less and less-plugin-clean-css:

```
$ sudo npm install -g less less-plugin-clean-css
```

on OS X, install nodejs via your preferred package manager (homebrew, macports) then install less and less-plugin-clean-css:

```
$ sudo npm install -g less less-plugin-clean-css
```

en Windows, instale nodejs , reinicie (para actualizar el PATH) e instale less y less-plugin-clean-css :

```
C:\> npm install -g less less-plugin-clean-css
```

Corriendo odoopath

Una vez que se configuran todas las dependencias, Odoopath se puede iniciar ejecutando odoopath.py.

La configuración se puede proporcionar a través de argumentos de la línea de comandos o a través de un archivo de configuración.

Las configuraciones necesarias comunes son:

```
PostgreSQL host, port, user and password.
```

Odoo no tiene valores predeterminados más allá de los valores predeterminados de psycopg2: se conecta a través de un socket UNIX en el puerto 5432 con el usuario actual y sin contraseña. Por defecto, esto debería funcionar en Linux y OS X, pero no funcionará en Windows, ya que no admite sockets de UNIX. Ruta de complementos personalizados más allá de los valores predeterminados, para cargar sus propios módulos

Bajo Windows una forma típica de ejecutar odoo sería:

```
C:\YourOdooPath> python odoo.py -w odoo -r odoo --addons-path=addons,../mymodules --db-filter=mydb$
```

Donde odoo, odoo son el nombre de usuario y la contraseña de postgresql, ../mymodules un directorio con complementos adicionales y mydb la base de datos predeterminada para servir en localhost: 8069

Bajo los sistemas * nix, una forma típica de ejecutar odoo sería:

```
$ ./odoo.py --addons-path=addons,../mymodules --db-filter=mydb$Packaged installers
```

¿Qué es Odoo?

Odoo (anteriormente conocido como OpenERP y antes de eso, TinyERP) es un conjunto de aplicaciones de administración empresarial de núcleo abierto. Dirigido a compañías de todos los tamaños, el conjunto de aplicaciones cubre todas las necesidades comerciales, desde el sitio web / comercio electrónico hasta la fabricación, el inventario y la contabilidad, todo integrado perfectamente. Es la primera vez que un editor de software logra alcanzar una cobertura tan funcional. Odoo es el software empresarial más instalado del mundo. Odoo es utilizado por más de 2,000,000 de usuarios en todo el mundo, desde compañías muy pequeñas (1 usuario) hasta compañías muy grandes (300,000 usuarios).

El código fuente para el marco de OpenObject y los módulos de ERP (planificación de recursos empresariales) centrales está a cargo de Odoo SA, con sede en Bélgica. Además, una comunidad global activa y una red de 500 socios oficiales proporcionan programación personalizada, soporte y otros servicios. Los componentes principales de Odoo son el marco de OpenObject, unos 30 módulos principales (también llamados módulos oficiales) y más de 3000 módulos comunitarios

Odoo ha sido utilizado como un componente de los cursos universitarios. Un estudio sobre aprendizaje experiencial sugirió que OpenERP proporciona una alternativa adecuada a los sistemas propietarios para complementar la enseñanza.

Se han escrito varios libros sobre Odoo, algunos de los cuales cubren áreas específicas como contabilidad o desarrollo.

Odoo ha recibido premios, incluidos los premios Trends Gazelle y BOSSIE durante tres años consecutivos.

Utiliza las secuencias de comandos de Python y PostgreSQL como su base de datos. Su edición comunitaria se complementa con una edición Enterprise @ USD 240 / - por usuario por año y una edición en línea con soporte comercial. El repositorio de desarrollo está en GitHub.

En 2013, se formó la Asociación de la Comunidad de Odoo sin fines de lucro para garantizar la promoción y el mantenimiento continuos de las versiones y módulos de la comunidad de Odoo para complementar el trabajo de Odoo SA Esta organización cuenta con más de 150 miembros que son una mezcla de personas y organizaciones.

Lea **Empezando con odoo-8 en línea**: <https://riptutorial.com/es/odoo-8/topic/2151/empezando-con-odoo-8>

Capítulo 2: ¿Cuáles son los métodos y detalles de ORM?

Observaciones

Método de creación: crea un nuevo registro con el valor especificado. Toma una serie de valores de campo y devuelve un conjunto de registros que contiene el registro creado

```
def create(self, vals):  
    return super(class_name, self).create(vals)
```

Método de escritura: actualice los registros con los identificadores dados con los valores de campo dados. Toma una serie de valores de campo, los escribe en todos los registros de su conjunto de registros. No devuelve nada

```
def write(self, vals):  
    return super(class_name, self).write(vals)
```

Método de búsqueda: busque registros basados en un dominio de búsqueda. Toma un dominio de búsqueda, devuelve un conjunto de registros de registros coincidentes. Puede devolver un subconjunto de registros coincidentes (parámetros de límite y desplazamiento) y ordenarse (parámetro de orden)

```
self.search([('customer', '=', True)])  
self.env['res.partner'].search([('partner', '=', True)])
```

Método de búsqueda: obtenga registros como objetos que permiten usar la notación de puntos para explorar campos y relaciones. Toma una identificación de la base de datos o una lista de identificaciones y devuelve un juego de registros, útil cuando las identificaciones de registros se obtienen desde fuera de Odoo (por ejemplo, ida y vuelta a través de un sistema externo) o al llamar a métodos en la antigua API.

```
self.browse([7, 8, 9])  
self.env['res.partner'].browse([7, 8, 9])
```

Métodos existentes: devuelve un nuevo conjunto de registros que contiene solo los registros que existen en la base de datos. Se puede usar para verificar si un registro (por ejemplo, obtenido externamente) todavía existe.

```
records = records.exists()
```

Método de referencia: método de entorno que devuelve el registro que coincide con un ID externo proporcionado

```
self.env.ref('base.group_public')
```

Método asegúrese de uno: comprueba que el conjunto de registros es un singleton (solo contiene un único registro), genera un error de lo contrario

```
records.ensure_one()
```

Examples

Diferentes tipos de métodos de ORM

1. crear()
2. escribir()
3. buscar()
4. vistazo()
5. existe ()
6. árbitro()
7. asegurar_one ()

Lea ¿Cuáles son los métodos y detalles de ORM? en línea: <https://riptutorial.com/es/odoo-8/topic/6150/-cuales-son-los-metodos-y-detalles-de-orm->

Capítulo 3: Añadir archivos CSS y Javascript al módulo Odoo

Sintaxis

- Nota sobre la sintaxis XML: como el registro se realiza dentro de un archivo XML, no puede dejar ninguna etiqueta sin cerrar como podría hacerlo en un HTML simple, como: `<link rel = 'hoja de estilo' href = "... " >` , Cierre el etiqueta de enlace en su lugar, como:
 - `<link rel = 'hoja de estilo' href = "... " / >`

Parámetros

Valores posibles del parámetro <i>inherit_id</i>	sentido
<i>web.assets_backend</i>	Usado solo en páginas internas, NO incluido en un sitio web público.
<i>website.assets_frontend</i>	Se utiliza solo en un sitio web público (a través del módulo " <i>sitio web</i> ").
<i>web.assets_common</i>	Utilizado tanto en sitio web público como en páginas internas.

Observaciones

Si no está seguro de qué opción es adecuada para usted, pruebe la primera opción (backend), ya que se utiliza en la mayoría de los casos y casi en todos los casos si no ha instalado el módulo "sitio web". Odoo distingue entre activos "backend" y "frontend" porque el sitio web público proporcionado por el módulo "website" utiliza un estilo y código JS diferente al de las páginas internas destinadas a tareas de ERP, es decir, "frontend" está asociado con el sitio web público y "backend" se asocia con páginas internas para ERP (el significado de "frontend" y "backend" son Odoo específicos aquí, pero ambos son "frontend" en sentido más general).

No solo puede elegir y usar una de las opciones, sino también usar cualquier combinación de ellas (dos de ellas o todas) en el mismo módulo. Factorice un backend, un frontend y un código JS / CSS común en archivos separados para adherirse mejor a DRY y tener el código adecuado en el sitio web público y en las páginas internas.

No olvide agregar "web" (cuando usa la *opción 1*) o "sitio web" (cuando usa la *opción 2*) a la lista de dependencias en el manifiesto `__openerp__.py` .

Examples

Almacena archivos CSS y JS correctamente en el módulo Odoo.

Los archivos CSS y JS deben residir bajo el directorio 'estático' en el directorio raíz del módulo (el resto del árbol de subdirectorios bajo 'estático' es una convención opcional):

- static / src / css / **your_file.css**
- static / src / js / **your_file.js**

Luego, agregue enlaces a estos archivos sin ver una de las 3 formas enumeradas en los siguientes ejemplos.

Opción 1: [BACKEND] Agrega archivos CSS y Javascript para usar en páginas internas

La forma Odoo v8.0 es agregar el registro correspondiente en el archivo XML:

- Agregue el archivo XML al manifiesto (es decir, el archivo `__openerp__.py` .):

```
...
'datos': ['tu_archivo.xml'],
...
```

- Luego agregue el siguiente registro en 'your_file.xml' :

```
<openerp>
  <data>
    <template id="assets_backend" name="your_module_name assets"
inherit_id="web.assets_backend">
      <xpath expr="." position="inside">
        <link rel='stylesheet' href="/your_module_name/static/src/css/
your_file.css" />
        <script type="text/javascript" src="/your_module_name/static/src/js/
your_file.js"></script>
      </xpath>
    </template>
    ....
    ....
  </data>
</openerp>
```

Opción 2: [FRONTEND] Agrega archivos CSS y Javascript para usar en un sitio web público

Nota: debe usar esta forma si ha instalado un módulo de "sitio web" y tiene un sitio web público disponible.

- Agregue el siguiente registro en 'your_file.xml' :

```

<openerp>
  <data>

    <template id="assets_frontend" name="your_module_name assets"
inherit_id="website.assets_frontend">
      <xpath expr="link[last()]" position="after">
        <link rel='stylesheet' href="/your_module_name/static/src/css/
your_file.css"/>
      </xpath>
      <xpath expr="script[last()]" position="after">
        <script type="text/javascript" src="/your_module_name/static/src/js/
your_file.js"></script>
      </xpath>
    </template>

  </data>
</openerp>

```

Opción 3: [COMÚN] Agregue archivos CSS y Javascript para usar en todas las páginas (backend y frontend)

- Agregue el siguiente registro en 'your_file.xml' :

```

<openerp>
  <data>

    <template id="assets_common" name="your_module_name assets"
inherit_id="web.assets_common">
      <xpath expr="." position="inside">
        <link rel='stylesheet' href="/your_module_name/static/src/css/
your_file.css"/>
        <script type="text/javascript" src="/your_module_name/static/src/js/
your_file.js"></script>
      </xpath>
    </template>

  </data>
</openerp>

```

Lea Añadir archivos CSS y Javascript al módulo Odoo en línea: <https://riptutorial.com/es/odoo-8/topic/3401/anadir-archivos-css-y-javascript-al-modulo-odoo>

Capítulo 4: Campos utilizados en Odoo 8

Introducción

Esta es la sección donde puede encontrar los detalles sobre los campos que se están utilizando en Odoo 8

Parámetros

Parámetros	Descripción
string = "Nombre"	Etiqueta opcional del campo.
compute = "_ compute_name_custom"	Transformar los campos en campos computados.
store = True	Si se calcula almacenará el resultado.
select = True	Índice de fuerza en campo
readonly = True	El campo será de solo lectura en vistas.
inverse = "_ write_name"	En la activación de actualización
requerido = Verdadero	Campo obligatorio
traducir = Verdadero	Habilitar la traducción
ayuda = 'blabla'	Texto de ayuda de ayuda
comodel_name = "model.name"	Nombre del modelo relacionado.
inverse_name = "field_name"	columna relacional del modelo opuesto
relación = 'many2many_table_name'	nombre de la tabla relacional para many2many
columns1 = 'left_column_name'	tabla relacional izquierda columna nombre
column2 = 'right_column_name'	tabla relacional derecha nombre de columna

Observaciones

Odoo y ORM: Odoo utiliza la técnica ORM (Mapeo relacional de objetos) para interactuar con la base de datos. ORM ayudará a crear una base de datos de objetos virtuales que se puede usar desde Python. En la técnica ORM, cada modelo está representado por una clase que subclasifica Models.model.

Models.model es la súper clase principal para los modelos de Odoo de base de datos regulares. Los modelos Odoo se crean heredando de esta clase.

Ejemplo:

```
class Employee(models.Model):
    _name = 'module.employee'

    #Rest of the code goes here
```

Aquí `_name` es un atributo estructural, que le dice al sistema sobre el nombre de la tabla de base de datos que se creará.

Cada modelo tiene una serie de variables de clase, cada una de las cuales representa un campo de base de datos en el modelo. Cada campo está representado por una instancia de una clase `openerp.fields.Field`. Los campos en Odoo se enumeran a continuación.

1 campo booleano

```
ex: flag = fields.Boolean()
```

2 Char Field

```
ex: flag = fields.Char()
```

3 texto

```
ex: flag = fields.Text()
```

4 html

```
ex: flag = fields.Html()
```

5 entero

```
ex: flag = fields.Integer()
```

6 flotador

```
ex: flag = fields.Float()
```

7 fecha

```
ex: flag = fields.Date()
```

8 fecha y hora

```
ex: flag = fields.Datetime()
```

9 Selección

```
ex: flag = fields.Selection()
```

10 Many2one

```
ex: flag = fields.Many2one()
```

11 One2many

```
ex: flag = fields.One2many()
```

12 Many2many

```
ex: flag = fields.Many2many()
```

Examples

Ejemplos de campos de Odoo 8

Odoo utiliza la técnica ORM (Object Relational Mapping) para interactuar con la base de datos. ORM ayudará a crear una base de datos de objetos virtuales que se puede usar desde Python. En la técnica ORM, cada modelo está representado por una clase que subclasifica `Models.model`. `Models.model` es la súper clase principal para los modelos de Odoo de base de datos regulares. Los modelos Odoo se crean heredando de esta clase.

```
name = fields.Char(string='New Value')

flag = fields.Boolean(string='Flag',default=False)

amount = fields.Float(string='Amount',digits=(32, 32))

code = fields.Selection(string='Code',selection=[('a', 'A'),('b', 'B')])

customer = fields.Many2one(comodel_name='res.users')

sale_order_line = fields.One2many(comodel_name='res.users', inverse_name='rel_id')

tags = fields.Many2many(comodel_name='res.users',
                        relation='table_name',
                        column1='col_name',
                        column2='other_col_name')
```

Lea Campos utilizados en Odoo 8 en línea: <https://riptutorial.com/es/odoo-8/topic/8152/campos-utilizados-en-odoo-8>

Capítulo 5: Cómo activar el modo de desarrollador OpenERP

Observaciones

Modo desarrollador

El modo de desarrollador de Odoo le permite realizar modificaciones sustanciales en la base de datos de Odoo, como agregar campos a sus documentos y vistas. Cambia las vistas predeterminadas de sus acciones e incluso puede crear formularios dinámicos basados en otros campos dentro de sus modelos.

Ventaja

Si bien Odoo es un marco de aplicación poderoso, el ciclo de desarrollo puede ser brutal para probar cambios en su aplicación. Al utilizar el modo de desarrollador, puede probar expresiones y resolver muchos problemas funcionales sin tener que reiniciar el servidor una y otra vez para probar cambios simples.

Además, la herramienta de desarrollo Odoo es excelente para observar la arquitectura de formas y vistas para ver cómo los campos están vinculados a los módulos, sus dominios, contextos y otros atributos. En este video exploramos exactamente cómo utilizamos estas herramientas para modificar y crear aplicaciones de Odoo.

Limitaciones

Si bien puede ser muy tentador utilizar el modo de desarrollador para realizar una gran cantidad de cambios en su aplicación, existen algunos inconvenientes. Dependiendo de lo que modifique y cambie, puede perder estos cambios con futuras actualizaciones del módulo o cuando instale aplicaciones adicionales en Odoo. Esto es particularmente cierto para los cambios en las vistas.

Para activar el modo de desarrollador simplemente escribe

para la versión v7

& debug =

antes de # firmar simplemente lo agregas.

http://localhost:8069/?db=test_db&debug=#

para la versión > v7

<http://localhost:8069/web?debug=>

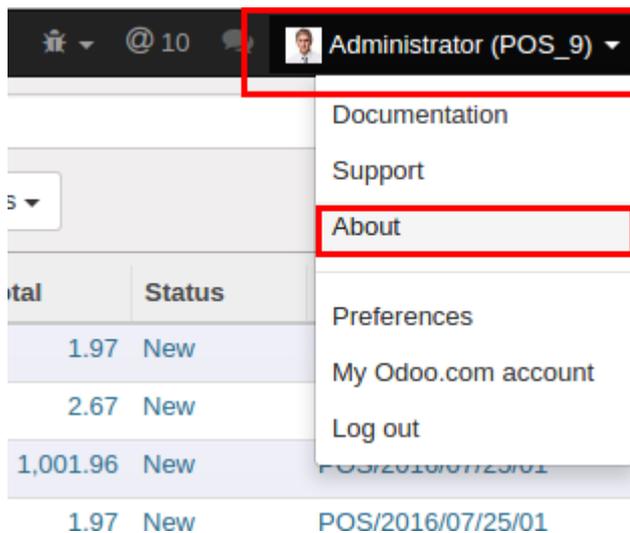
Es posible que no vea el menú **Acerca de Odoo** porque es posible que **haya** instalado el **módulo de marca de datos de Odoo** .

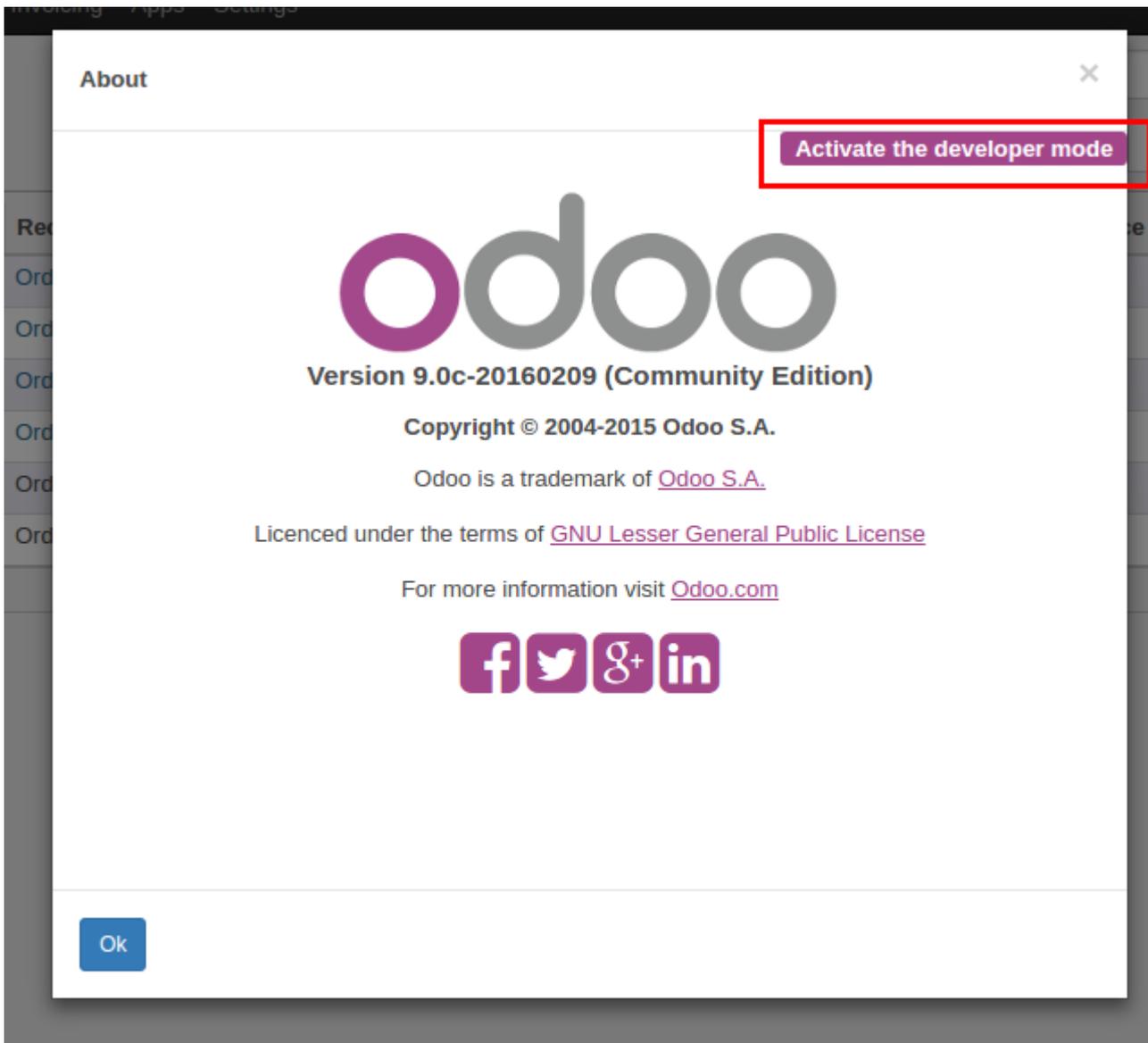
Examples

Activar el modo desarrollador

Para activar el modo desarrollador:

1. Inicie sesión en la parte frontal de ODOO
2. Haga clic en el nombre de usuario desplegable en la parte superior derecha
3. Seleccione 'Acerca de'
4. Haga clic en 'Activar modo de desarrollador' en la ventana emergente.





Activando el modo desarrollador en Odoo 8

Cuando inicie sesión en la aplicación Odoo, encontrará una opción para ver quién es la persona que ha iniciado sesión en la esquina superior derecha. Esta información de usuario tiene un botón desplegable. Haga clic en el menú desplegable, luego encontrará una lista. En esa lista seleccione sobre la opción Odoo.com. Al hacer clic en eso se abrirá una ventana emergente **Acerca de** . En esa ventana, en la esquina superior derecha encontrará una opción como **Activar el modo de desarrollador** . Al hacer clic en ese enlace se volverá a cargar la página web.

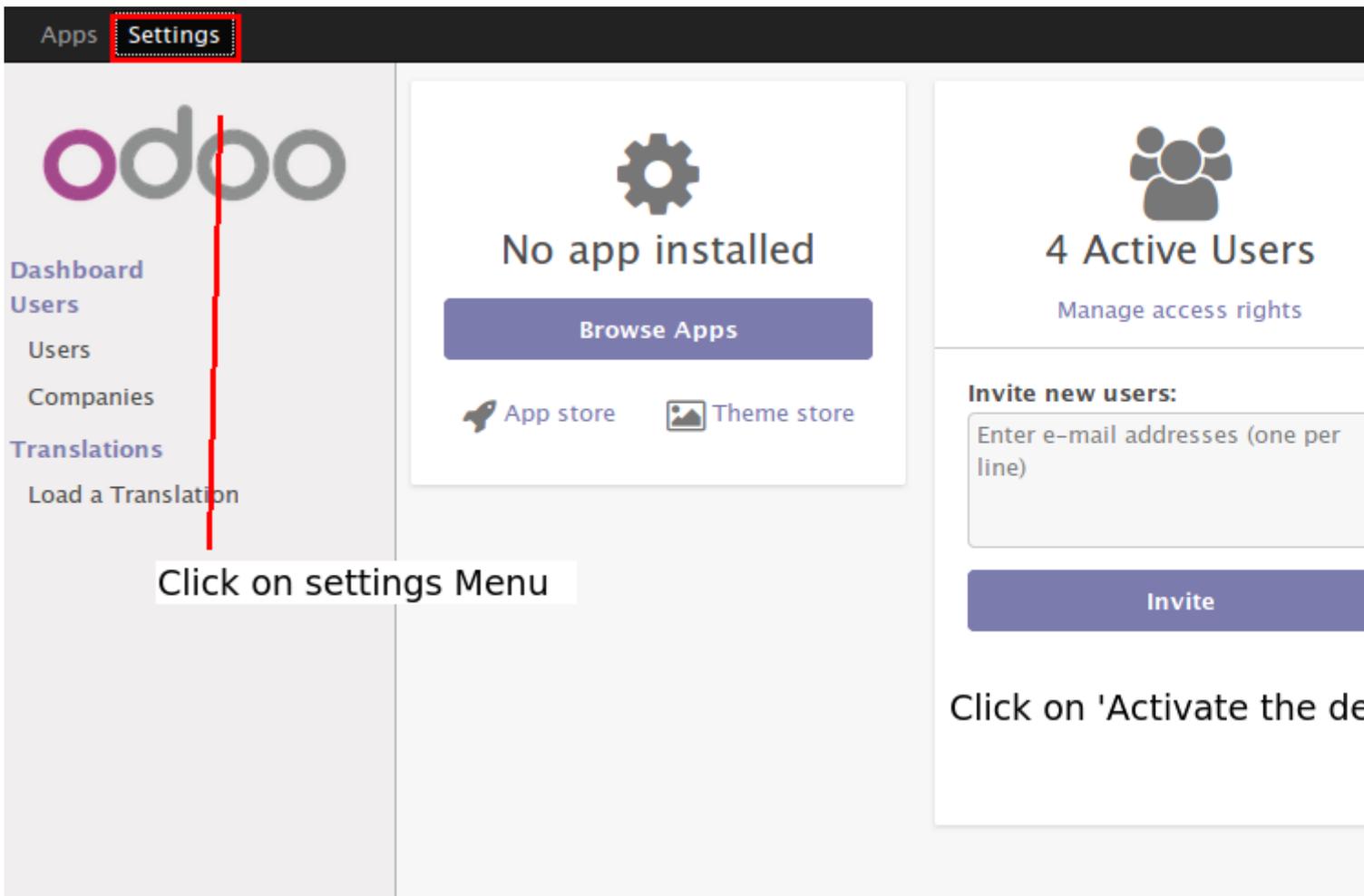
Después de volver a cargar, estará en modo desarrollador. Luego, el enlace cambiará a algo como esto http://localhost:8069/web?Debug=#id=23&view_type=form&model=res.partner

Activar el modo desarrollador en Odoo 10

Activar el modo de desarrollador:

1. Inicie sesión en la aplicación odoo.

2. Después de iniciar sesión el usuario puede ver varios menús odoo. Haga clic en el menú de configuración.



1. Haga clic en 'Activar el modo de desarrollador' que se encuentra en la esquina inferior derecha de la página de configuración.
2. Modo de desarrollador ahora activado.



Lea Cómo activar el modo de desarrollador OpenERP en línea: <https://riptutorial.com/es/odoo-8/topic/3311/como-activar-el-modo-de-desarrollador-openerp>

Capítulo 6: Configurar correo electrónico - Office 365 en Odoo

Examples

Configurar E-Mail

- Inicialmente revise su configuración de correo electrónico

POP and IMAP settings

Use the information on this page if you need to use POP or IMAP to connect your mailbox.

POP setting

Server name: outlook.office365.com

Port: 995

Encryption method: SSL

IMAP setting

Server name: outlook.office365.com

Port: 993

Encryption method: SSL

SMTP setting

Server name: smtp.office365.com

Port: 587

Encryption method: TLS

POP options

- Send event invitations in iCalendar format
- Don't send receipts for messages that have been read

IMAP options

- Send event invitations in iCalendar format
- Don't send receipts for messages that have been read

-
- **En Odoo vaya a Configuración -> Correo electrónico.**

The screenshot displays the OpenERP web interface. The browser's address bar shows the URL: `127.0.0.1:8069/?db=geserpupdated&ts=1448886099885#id=1&view_type=form&mo`. The left sidebar menu is circled in red, with the following items visible: Sales, Warehouse, Project, Accounting, Human Resources, General Settings, Companies, Users, Import Module, Translations, and Technical. Under the 'Technical' category, the 'Email' subcategory is expanded, and 'Incoming Mail Servers' is highlighted. The main content area has two tabs: 'Server & Login' (selected) and 'Advanced'. The 'Server Information' section contains the following data:

Server Name	outlook.office365.com
Port	995
SSL/TLS	<input checked="" type="checkbox"/>

Below the server information, there is a section titled 'Actions to Perform on Incoming Mails' with a button labeled 'Create a New Record'.

- **Ingrese los valores de campo en "Opciones de servidores de correo entrante" y "Servidores de correo saliente".**

Name

Gmail

Server Type

POP Server

Last Fetch Date

Server & Login

Advanced

Server Information

Server Name

outlook.office365.com

Port

995

SSL/TLS



Login Information

Username

Password

Actions to Perform on Incoming Mails

Create a New Record

Server Action

Description

Office

Priority

10

Connection Information

SMTP Server

outlook.office365.com

SMTP Port

25

Debugging



Security and Authentication

Connection Security

Username

Password

TLS (STARTTLS)

 Test Connection

Lea Configurar correo electrónico - Office 365 en Odoo en línea: <https://riptutorial.com/es/odoo-8/topic/6648/configurar-correo-electronico---office-365-en-odoo>

Capítulo 7: Crear funciones automatizadas para el modelo

Introducción

A menudo necesitamos ejecutar algún código automáticamente durante la instalación del módulo. Esto tiene muchas razones, por ejemplo, la configuración de los módulos de `sale` para cumplir con los requisitos de nuestro proyecto.

En este tema, aprenderá cómo hacer que la función automatizada se ejecute en la instalación del módulo.

Examples

En primer lugar, necesita crear un archivo xml para realizar una llamada a la función

```
<?xml version="1.0"?>
<openerp>
  <data noupdate="1">
    <function model="*model_name*" name="_configure_sales"/>
  </data>
</openerp>
```

Este archivo XML es sencillo llamadas `_configure_sales` función del modelo *nombre_del_modelo*.

NOTA: este archivo xml debe estar en la parte superior de `data` matriz de `data`, porque Odoo está procesando archivos xml de arriba a abajo.

Archivo Python correspondiente

```
class *model_name*(models.Model):
    _name = *model_name*

    @api.model
    def _configure_sales(self):
        # Do the configuration here
```

Cada vez que se instale el módulo se ejecutará esta función.

Nota: Si elimina `noupdate` de xml, la función también se ejecutará en la actualización.

Lea Crear funciones automatizadas para el modelo en línea: <https://riptutorial.com/es/odoo-8/topic/10633/crear-funciones-automatizadas-para-el-modelo>

Capítulo 8: RPC utilizando la API de Odoo v8 (función Call Python desde JavaScript)

Observaciones

Si está considerando agregar nuevos métodos en Python para usarlos en RPC desde JavaScript, considere las siguientes opciones de decoradores de métodos: si tiene que lidiar con identificadores / conjuntos de registros, entonces para la definición del método python elija decorador:

- `@ api.multi` - para obtener un *conjunto de registros* en su método
- `@ api.one`: para obtener *browse_records* uno por uno en su método en los ejemplos anteriores se usa `@ api.multi`, pero `@ api.one` también se puede usar para tratar los identificadores, dependiendo de los requisitos (Sin embargo, se recomienda encarecidamente usar `@ api.multi` en lugar de `@ api.one` por razones de rendimiento).

O si es una función simple que no tiene que lidiar con registros / ID, entonces para el método python elige decorador:

- `@ api.model` - Permite ser educado con la API de estilo antiguo.
- `@ api.multi` - Nuevamente, también puede usarlo aquí, simplemente pase `[]` (matriz vacía) como primer argumento en javascript ...

Referencias: [documentación de Odoo RPC](#) , [decoradores de métodos Odoo 8 API](#)

Examples

Un ejemplo de modelo de Odoo para llamar a métodos de

```
class my_model(models.Model):
    _name = "my.model"

    name = fields.Char('Name')

    @api.multi
    def foo_manipulate_records_1(self):
        """ function returns list of tuples (id,name) """
        return [(i.id,i.name) for i in self]

    @api.multi
    def foo_manipulate_records_2(self, arg1, arg2):
        #here you can take advantage of "self" recordset and same time use additional arguments
        "arg1", "arg2"
        pass

    @api.model
    def bar_no_deal_with_ids(self, arg1, arg2):
        """ concatenate arg1 and arg2 """
```

```
return unicode(arg1) + unicode(arg2)
```

Ejemplos de Odoo RPC

Los siguientes ejemplos muestran cómo llamar a la función Python desde JavaScript en Odoo 8. En los ejemplos, llamamos a los métodos de *my_model* descritos anteriormente en esta página. Suponemos que en los siguientes ejemplos, la variable "list_of_ids" contiene una lista (matriz) de identificadores de registros existentes del modelo "my.model".

- Llamada del método **foo_manipulate_records_1** decorado con **@ api.multi** :

```
new instance.web.Model("my.model")
  .call( "foo_manipulate_records_1", [list_of_ids])
  .then(function (result) {
    // do something with result
  });
```

- Llamada del método **foo_manipulate_records_2** decorado con **@ api.multi** :

```
new instance.web.Model("my.model")
  .call( "foo_manipulate_records_2", [list_of_ids, arg1, arg2])
  .then(function (result) {
    // do something with result
  });
```

- Llamada del método **bar_no_deal_with_ids** decorado con **@ api.model** :

```
new instance.web.Model("my.model")
  .call( "bar_no_deal_with_ids", [arg1, arg2])
  .then(function (result) {
    // do something with result
  });
```

Además, si tiene algún sentido dependiendo de la implementación, puede llamar a la función decorada con **@ api.multi** incluso si no tiene que lidiar con identificadores (simplemente pase una matriz vacía en lugar de identificadores, como primer elemento de la lista de argumentos):

```
new instance.web.Model("my.model")
  .call( "foo_manipulate_records_2", [[], arg1, arg2])
  .then(function (result) {
    // do something with result
  });
```

de esta manera puede ser útil en algunos casos, ya que la función no decorada en la vi.0 v8.0 se considera como **@ api.multi** (ya que **@ api.multi** es un decorador predeterminado)

Puede utilizar el **tercer parámetro** , un **diccionario de argumentos de palabras clave**, a partir de dos parámetros para llamadas RPC que se utilizan en los ejemplos anteriores (el nombre de la función y la lista de argumentos). Es muy recomendable cambiar el contexto (en algunos casos

puede ser incluso necesario), ya que puede cambiar el comportamiento del procedimiento remoto (localización, etc.). Vea a continuación el ejemplo con argumento de contexto en la llamada RPC (se puede aplicar lo mismo a todos los ejemplos anteriores)

```
var self = this;
new instance.web.Model("my.model")
    .call("foo_manipulate_records_2", [], arg1, arg2], {'context':self.session.user_context})
    .then(function (result) {
        // do something with result
    });
```

Por supuesto, también puede usar un contexto personalizado, si es necesario, en lugar de cambiar el existente como en este ejemplo.

Lea [RPC utilizando la API de Odoo v8 \(función Call Python desde JavaScript\) en línea:](https://riptutorial.com/es/odoo-8/topic/6613/rpc-utilizando-la-api-de-odoo-v8--funcion-call-python-desde-javascript-)
<https://riptutorial.com/es/odoo-8/topic/6613/rpc-utilizando-la-api-de-odoo-v8--funcion-call-python-desde-javascript->

Capítulo 9: Widgets personalizados para campos

Observaciones

- Asegúrese de [agregar](#) correctamente el [archivo javascript a su módulo](#)
- no olvide agregar 'web' como dependencia en `__openerp__.py`:

```
'depends': ['web', ...]
```

Examples

Widget personalizado para campos numéricos para usar en TreeView

El siguiente ejemplo de widget muestra cómo formatear celdas individuales de una columna TreeView condicionalmente, dependiendo del valor del campo en la celda particular. Si el valor del campo es negativo, entonces se mostrará en color rojo y el símbolo menos se ocultará, de lo contrario se mostrará en color normal.

Un widget debe estar escrito en JavaScript, vamos a usar `custom_widget_name` como nombre para un nuevo widget, y `your_module_name` es un nombre técnico de su módulo (igual que el nombre del directorio raíz de su módulo)

En la carpeta estática `/src/js/` en su módulo, agregue el archivo javascript (por ejemplo, `static/src/js/custom_widget.js`) con un widget personalizado:

```
openerp.your_module_name = function (instance) {

    instance.web.list.columns.add('field.custom_widget_name',
    'instance.your_module_name.custom_widget_name');

    instance.your_module_name.custom_widget_name = instance.web.list.Column.extend({
        _format: function (row_data, options) {
            res = this._super.apply(this, arguments);
            var amount = parseFloat(res);
            if (amount < 0){
                return "<font color='#ff0000'>" + (-amount) + "</font>";
            }
            return res
        }
    });
    //
    //here you can add more widgets if you need, as above...
    //
};
```

el widget de ejemplo anterior se puede usar en una vista de lista para el campo de tipo float y aplica reglas personalizadas de la siguiente manera:

- Números negativos:
 - Se muestran en rojo.
 - El símbolo menos (un carácter '-') está "oculto".
- Para números positivos se utiliza el diseño por defecto.

Este widget de ejemplo se puede aplicar a un campo en una vista de árbol de Odoo. Puede usar un widget como este para una columna a la que debe aplicar las reglas personalizadas para:

```
. . .  
<tree >  
  . . .  
  <field name="some_field" widget="my_widget" />  
  . . .  
</tree>  
. . .
```

Lea Widgets personalizados para campos en línea: <https://riptutorial.com/es/odoo-8/topic/6198/widgets-personalizados-para-campos>

Creditos

S. No	Capítulos	Contributors
1	Empezando con odoo-8	4444 , Community , danidee , T.V.
2	¿Cuáles son los métodos y detalles de ORM?	AKHIL MATHEW , Mani
3	Añadir archivos CSS y Javascript al módulo Odoo	George Daramouskas , T.V.
4	Campos utilizados en Odoo 8	AKHIL MATHEW
5	Cómo activar el modo de desarrollador OpenERP	AKHIL MATHEW , Emipro Technologies Pvt. Ltd. , Gopakumar N G , Mehedi Hasan
6	Configurar correo electrónico - Office 365 en Odoo	Don Chakkappan
7	Crear funciones automatizadas para el modelo	Dachi Darchiashvili
8	RPC utilizando la API de Odoo v8 (función Call Python desde JavaScript)	T.V.
9	Widgets personalizados para campos	T.V.