



链滴

# windows/ubuntu16/centos7 安装 odoo1 1

作者: [AlanSune](#)

原文链接: <https://ld246.com/article/1591526696750>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



本文介绍odoo11在windows, ubuntu16, centos7下的安装，涉及安装中的依赖处理和各种工具的装。

## 64位windows系统环境下的安装

### 1. 安装python3.6

安装程序下载地址：<https://www.python.org/downloads/>

安装后在cmd中输入 `python --version` 和 `pip3 --version` 查看版本信息。如果提示命令不存在，请将 `python3.6` 和 `python3.6/scripts` 文件夹路径添加到系统路径。

注意：如果之前已经安装python2，请将python3.6安装目录下的`python.exe`复制，并将副本名称修为`python3.exe`，之后使用`python3`来调用。

### 2. 安装git

安装程序下载地址：<https://git-scm.com/>

安装后在cmd中输入`git`查看是否安装成功。

### 3. 安装依赖：

lxml依赖： `lxml-4.1.1-cp36-cp36m-win_amd64.whl`

pillow依赖： `Pillow-3.4.2-cp36-cp36m-win_amd64.whl`

pypiwin32依赖： `pypiwin32-220-cp36-none-win_amd64.whl`

greenlet依赖： `greenlet-0.4.12-cp36-cp36m-win_amd64.whl`

pyldap依赖： `pyldap-2.4.45-cp36-cp36m-win_amd64.whl`

上述whl包使用 `pip3 install` + 文件名安装即可。 (上述包可以在网络上查找, 查找不到请联系我)

4.安装postgresql数据库, 创建用户, 要拥有创建数据库的权限。

5.下载源码:

在cmd中执行以下命令下载源码。

```
git clone https://www.github.com/odoo/odoo --depth 1 --branch 11.0
```

6.修改源码中的requirements.txt文件:

将pyldap==2.4.28删除 (或在前面添加#注释)

将lxml==3.5.0, Pillow==3.4.1, greenlet==0.4.10, Werkzeug==0.11.11

修改为lxml==4.1.1, Pillow==3.4.2, greenlet==0.4.12, Werkzeug==0.11.15

创建文件 %HOMEPATH%/pip/pip.ini, (已存在修改即可) , 修改pip.ini为

```
[global]
```

```
trusted-host = mirrors.ustc.edu.cn
```

```
index-url = https://mirrors.ustc.edu.cn/pypi/web/simple
```

在项目根目录下执行 `pip3 install -r requirements.txt`, 执行成功即可。

7.添加配置文件odoo.conf, 参考odoo9数据库配置, 运行

```
python3 odoo-bin -c odoo.conf
```

按照上述步骤安装的依赖会直接安装到python3的库中, 建议使用虚拟环境安装 (可用工具: pipenv, 参考Ubuntu 16.04环境下的安装 (第7步))

## Ubuntu 16.04环境下的安装

参考地址: <https://webkul.com/blog/install-odoo11-on-ubuntu/>

1.安装虚拟机, 并在虚拟机中安装Ubuntu 16.04系统。Ubuntu 16.04默认安装了python3和python (使用`python3 -V` 和 `python -V`查看具体版本)。

2.更新apt源:

```
sudo apt update
```

3.安装postgresql:

```
sudo apt install -y postgresql
```

启动服务, 创建postgresql用户:

```
sudo service postgresql start
```

```
sudo su - postgres
```

```
createuser --createdb --username postgres --no-createrole --no-superuser --pwprompt odoo  
exit
```

修改用户连接方式:

```
sudo vi /etc/postgresql/9.5/main/pg_hba.conf
```

修改以下内容:

```
# "local" is for Unix domain socket connections only
```

```
local all all peer md5
```

然后重启服务:

```
sudo service postgresql restart
```

4.安装wkhtmltopdf:

```
sudo wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.1/wkhtmltopdf-0.12.1_linux-trusty-amd64.deb
```

```
sudo dpkg -i wkhtmltopdf-0.12.1_linux-trusty-amd64.deb
```

```
sudo cp /usr/local/bin/wkhtmltopdf /usr/bin
```

```
sudo cp /usr/local/bin/wkhtmltoimage /usr/bin
```

安装其他环境:

```
sudo apt install python-dev libsasl2-dev libldap2-dev libssl-dev  
sudo apt install libxml2-dev libxslt-dev
```

5.安装pip3:

```
sudo apt install -y python3-pip
```

安装git

```
sudo apt install -y git
```

6.安装nodejs和less

```
sudo apt-get install -y npm
```

```
sudo ln -s /usr/bin/nodejs /usr/bin/node
```

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

```
sudo apt-get install -y node-less
```

7.下载源码（请注意下载目录）：

```
git clone https://www.github.com/odoo/odoo --depth 1 --branch 11.0
```

切换到源码目录，安装pipenv,创建虚拟环境，并安装依赖：

```
sudo pip3 install pipenv
```

```
pipenv --python 3.5 install -r requirements.txt
```

## 8.配置启动文件和启动

在源码路径下创建文件odoo.conf，添加内容并修改对应内容：

```
[options]
```

;模块路径

```
addons_path = odoo addons, addons
```

;超级管理员密码

```
admin_passwd = admin
```

```
db_host = localhost
```

```
db_port = 5432
```

```
db_maxconn = 64
```

;数据库名称

```
db_name = odoo
```

;数据库用户

```
db_user = odoo
```

;数据库密码

```
db_password = 123456
```

```
db_sslmode = prefer
```

```
db_template = template1
```

在终端中执行

```
pipenv run python3 odoo-bin -c odoo.conf
```

访问网址:<http://localhost:8069>

## CentOS7环境下的安装

### 1.安装epel-release等相关依赖

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

```
sudo yum -y update
```

### 2.安装nodejs和less插件

```
sudo yum install -y nodejs
```

### 3.安装postgresql:

```
# 安装
sudo yum install -y postgresql-server

# 初始化
service postgresql initdb

# 查看服务状态
systemctl status postgresql

# 启动服务
systemctl start postgresql

# 开机运行服务
systemctl enable postgresql

# 切换到数据库用户
su - postgres

# 使用psql命令来进入PostgreSQL
psql

# 需要创建postgres密码
\password postgres

# 输入2次密码

# 退出
\q

# 创建odoo使用用户，此处为odoo，并输入密码
createuser --createdb --username postgres --no-createrole --no-superuser --pwprompt odoo

# 退出数据库用户
exit

# 修改用户连接属性
vim /var/lib/pgsql/data/pg_hba.conf

修改以下内容:
```

```
local all all peer md5
```

保存，重启数据库

```
systemctl restart postgresql
```

4.安装wkhtmltopdf:

```
sudo wget https://github.com/wkhtmltopdf/wkhtmltopdf/releases/download/0.12.2.1/wkhtml  
ox-0.12.2.1_linux-centos7-amd64.rpm
```

```
sudo yum install -y xorg-x11-fonts-Type1 xorg-x11-fonts-75dpi
```

```
sudo rpm -i wkhtmltox-0.12.2.1_linux-centos7-amd64.rpm
```

5.安装python3.6:

```
#使用 rpm 包进行安装安装python3.6
```

```
yum -y install https://centos7.iuscommunity.org/ius-release.rpm
```

```
yum makecache
```

```
yum install python36u
```

```
yum -y install python36u-pip
```

```
yum -y install python36u-devel
```

```
#查找python3.6安装位置
```

```
which python3.6
```

```
#得到位置 /usr/bin/python3.6, 新增python3的链接
```

```
ln -s /usr/bin/python3.6 /usr/bin/python3
```

```
#查找pip3安装位置
```

```
which python3.6
```

```
#得到位置 /usr/bin/pip3.6, 新增pip3的链接
```

```
ln -s /usr/bin/pip3.6 /usr/bin/pip3
```

6.安装git

```
sudo yum install -y git
```

7.安装odoo

```
#安装依赖, 第一行pyldap的依赖, 第二行为lxml的依赖
```

```
sudo yum install python-devel openldap-devel
```

```
yum install libxml2-devel libxml++-devel libxml2-devel  
#下载源码（请注意下载目录），此处如果使用公司odoo11的源码，注意源码目录结构  
git clone https://www.github.com/odoo/odoo --depth 1 --branch 11.0  
#删除window下的依赖,所有带有 sys_platform == 'win32' 的行删除，并保存  
vim requirements.txt  
#切换到源码目录，安装pipenv,创建虚拟环境  
sudo pip3 install pipenv  
pipenv --python 3.6  
#修改pipenv的url并保存  
vim Pipfile  
url = "https://pypi.python.org/simple"  
url = "https://pypi.tuna.tsinghua.edu.cn/simple"  
#下载依赖  
pipenv install  
在源码路径下创建文件odoo.conf，添加以下红色内容  
[options]  
;模块路径  
addons_path = odoo/addons, addons  
;超级管理员密码  
admin_passwd = admin  
;数据库IP  
db_host = localhost  
;数据库端口  
db_port = 5432  
db_maxconn = 64  
;数据库名称  
db_name = odoo  
;数据库用户  
db_user = odoo  
;数据库密码
```

```
db_password = 123456
db_sslmode = prefer
db_template = template1
```

保存后，在终端中执行

```
pipenv run python3 odoo-bin -c odoo.conf
```

访问网址<http://centos7的IP:8069>

## 与odoo 9相比的变化

1. 开发使用python3，官方文档使用python3.5+，请按照安装步骤安装python3.6后再进行开发。

2. 底层部分模型发生了改变，需要升级数据库才能直接使用odoo11。官方针对odoo企业版提供数据升级的功能，目前暂无odoo11的升级方案，社区版不提供升级。

3. odoo11移除了工作流，无法使用工作流进行开发。

如果迁移代码，移除工作流，修改以下内容：

修改view中的类型为 `workflow` 的 `button`，修改类型为 `object`，并触发工作流下一步触发的方法。

修改python代码中使用 `signal_workflow` 触发工作流的方法，使用直接调用工作流下一步触发的方法。

删除工作流文件，并删除 `openerp.py` 中配置的文件路径。

如果自己添加工作流，需要修改以下部分（参考部分使用odoo10），修改工作量较大，而且修改底层模型，无法估计其他影响：

添加工作流模型，参考 `odoo/addons/base/workflow`

修改工作流模型权限，参考 `odoo/addons/base/security`

添加工作流的处理对象，参考 `odoo/workflow`

修改基础模型，增加模型变化时对应的工作流变化，参考 `odoo/models.BaseModel`

增加js中对button按钮类型的处理(`addons\web\static\src\js\chrome\view_manager.js`)，参考 `addons\web\static\src\js\framework\view.js`

增加js中调用后台的rpc方法(`addons\web\static\src\js\_deprecated\data.js`)，参考 `addons\web\static\src\js\framework\model.js`

修改web路由，添加到工作流的路由方法，参考 `addons.DataSet.exec_workflow`

```
@http.route('/web/dataset/exec_workflow', type='json', auth="user")
```

4. odoo11的的定时器修改为使用 `ir.actions.server` 来触发任务，所以在创建定时器的时候需要修改以内容：

```
<record id="ir_cron_check_challenge" model="ir.cron">
    <field name="name">Gamification: Goal Challenge Check</field>
    <field name="model_id" ref="model_gamification_challenge"/> #使用model_id替换model_id，实际为ir.actions.server模型中的model_id
```

```

<field name="state">code</field> #ir.actions.server使用代码执行，其他
选项参考ir.actions.server
<field name="code">model._cron_update()</field> #ir.actions.server执行代码
<field name="interval_number">1</field>
<field name="interval_type">days</field>
<field name="numbercall">-1</field>
<field eval="False" name="doall" />
</record>

```

也可以修改 `ir.cron` 的 `create` 方法来适配（可能引起其他问题，应当每个模块修改）：

```

@api.model
def create(self, values):
    values['usage'] = 'ir_cron'
    # 添加新旧数据转化
    if values.get('model'):
        module = self._context.get('install_mode_data').get('module')
        model_xml_id = (module + '.' if module else '') + 'model_' + values['model'].replace('.', '_')
        values['model_id'] = self.env.ref(model_xml_id).id
        values['state'] = 'code'
        del values['model']
    if values.get('function') and values.get('args'):
        values['code'] = 'model.' + values['function'] + values['args']
        del values['function']
        del values['args']
    return super(ir_cron, self).create(values)

```

5.odoo11将 `addons/report` 中的打印模板移动到了 `addons/web`,故所有打印的模板应该使用 `web.html_container` 和 `web.external_layout` 来替换 `report.html_container` 和 `report.external_layout`. 请查模块是否使用打印模板。

6.odoo11禁止使用旧的API，所有使用旧API的方法请进行修改。最主要的例子为获取系统url的方法。

```

def get_base_url(self, cr, uid):
    base_url = self.pool.get('ir.config_parameter').get_param(cr, uid, 'web.base.url')
    return base_url

```

修改为：

```

def get_base_url(self):
    base_url = self.env['ir.config_parameter'].get_param('web.base.url')
    return base_url

```

7.视图中的 `kanban` 视图如果使用 `record.id.value`，如下：

```

```

请添加 `<field name="id"/>`，否则odoo11无法识别id字段。

前端框架中 `web.rpc` 代替 `web.Model`，前端修改参考：<https://www.youtube.com/watch?v=u-6aL1oqcw>

8.odoo11中不再使用 `openerp`,全部改为 `odo`。

`openerp` 虽然可以正常使用，但对于调试和后续升级带来影响，建议全部修改。以下为部分举例：

python代码中使用

```
from odoo import models, fields, api
```

替换

```
from openerp import models, fields, api
```

xml文件中使用 `<odoo></odoo>` 替换 `<openerp></openerp>` (参考odoo官方文档data files模，odoo8和以后版本的区别)

9.其他已知修改，影响未知：

hr模块移除 `last_login` 字段，注意视图修改。

ir.values模块移除，新增 ir.defaults 模块。

ir.ui.menu模型从odoo10开始增加 active 字段。

新增 `odoo/addons/base/module/wizard/base_module_uninstall`, 去除 `base_module_configuration`。

ir\_act\_client, ir\_act\_report\_xml, ir\_act\_drtver, ir\_act\_url, ir\_act\_window, ir\_action 模型进行了修。  
。

新增 web.tour 模块。

新增 addons/iap。