

Odoo 10 开发概要

(英文丛书原名 : Odoo10 Development Essentials)

快速更新你的开发技能 , 构建强大的Odoo 10业务应用程序

Daniel Reis 著 上海-老杨 (杨浚波) 译

QQ : 291525012

E-mail : yangxunbo@qq.com

Wechat : yangxunbo1986

Odoo中文社区 (odoo.net.cn) -内部开发文档系列

感谢以下真正的开源勇士单位的支持 :



版权申明: 本丛书一律免费共享, 未经授权同意切勿商业应用, 违者追究法律责任!

PS：如果此书对您有很大的帮助，您也乐意为开源事业做出贡献请支持译者，扫描下图二维码转账支持译者：



打开支付宝[扫一扫]



杨浔波

免费寄送收钱码：拨打95188-6

优秀靠谱的odoo服务及咨询公司名单列表：

上海先安信息科技有限公司

上海开阖软件有限公司

上海龙坤信息科技有限公司

上海寰享网络科技有限公司

上海卓忆科技发展有限公司

上海开石信息技术有限公司

上海巨盈信息技术有限公司

青岛欧度软件技术有限责任公司

前言

Odoo是一个强大的商业应用开源平台。在此基础上，构建了一套紧密集成的应用程序，涵盖了从CRM到销售到股票和会计的所有业务领域。Odoo有一个动态和不断增长的社区，不断增加功能、连接器和其他商业应用。

Odoo 10开发要点提供了一个逐步指导Odoo开发的指南，让读者能够快速的爬上学习曲线，并在Odoo应用平台上变得富有成效。

前两章的目的是让读者熟悉Odoo，学习建立开发环境的基本技术，熟悉模块开发方法和工作流。

以下各章节详细解释了Odoo addon模块开发所需的关键开发主题，如继承和扩展、数据文件、模型、视图、业务逻辑等等。

最后，最后一章解释了在部署Odoo实例时应该考虑什么。

What this book covers

第1章，开始了Odoo开发，从开发环境的设置开始，从源代码安装Odoo，并学习如何管理Odoo服务器实例。

第2章，构建您的第一个Odoo应用程序，指导我们创建第一个Odoo模块，涵盖涉及的所有不同层:模型、视图和业务逻辑。

第3章，继承——扩展现有的应用程序，解释现有的继承机制，以及如何使用它们创建扩展模块，在其他现有模块上添加或修改功能。

第4章，模块数据，包括最常用的Odoo数据文件格式(XML和CSV)，外部标识符概念，以及如何在模块和数据导入/导出中使用数据文件。

第5章，模型构建应用程序数据，详细讨论模型层，使用模型和字段的类型，包括关系和计算字段。

第6章，视图——设计用户界面，包括视图层，详细解释了几种类型的视图以及可以用来创建动态和直观的用户界面的所有元素。

第7章，ORM应用程序逻辑——支持业务流程，在服务器端引入编程业务逻辑，探索ORM概念和特性，并解释如何使用向导进行更复杂的用户交互。

第8章，编写测试和调试代码，讨论如何向addon模块添加自动化测试，以及调试模块业务逻辑的技术。

第9章，QWeb和看板视图，通过Odoo QWeb模板，使用它创建丰富的看板。

第10章，创建QWeb报告，讨论使用基于QWeb的报告引擎，以及生成友好的PDF报告所需要的一切。

第11章，创建网站前端功能，介绍了Odoo网站开发，包括web控制器实现和使用QWeb模板构建前端web页面。

第12章，外部API——与其他系统集成，解释了如何从外部应用程序中使用Odoo服务器逻辑，并引入了一个受欢迎的客户端编程库，也可以作为命令行客户端使用。

第13章，部署清单——现场直播，向我们展示了如何为生产黄金时间准备一个服务器，解释应该注意哪些配置，以及如何配置Nginx反向代理以提高安全性和可伸缩性。

What you need for this book

我们将在Ubuntu或Debian系统上安装我们的Odoo服务器，但我们希望您使用您的操作系统和编程工具，无论是Windows、Mac还是其他。

我们将提供一些关于在Ubuntu服务器上设置虚拟机的指导。您应该选择使用的虚拟化软件，例如VirtualBox或VMWare Player;两者都是免费的。如果您使用的是Ubuntu或Debian工作站，则不需要虚拟机。

正如您已经指出的，我们的Odoo安装将使用Linux，因此我们将不可避免地使用命令行。但是，你应该能够按照所给的指令行事，即使不熟悉它。

预期Python编程语言的基本知识。如果你不喜欢它，我们建议你学习快速教程，让你开始。我们还将使用XML，因此我们希望熟悉标记语法。

Who this book is for

这本书的目标是开发人员，他们有开发商业应用程序的经验，他们愿意快速用Odoo来生产。

读者应该了解MVC应用程序设计和Python编程语言的知识。熟悉web技术、HTML、CSS和JavaScript也会有所帮助。

Conventions

在这本书中，你会发现许多不同种类的信息的文本样式。以下是这些风格的一些例子，以及它们的含义的解释。

文本中的代码单词、数据库表名、文件夹名称、文件名、文件扩展名、路径名、虚拟url、用户输入和Twitter句柄如下：文本中的代码字如下所示：“创建一个新的数据库，使用createdb命令。”

代码块设置如下：

```
@api.multi
def do_toggle_done(self):
    for task in self:
        task.is_done = not task.is_done
    return True
```

当我们将您的注意力吸引到代码块的某个特定部分时，相关的行或项以粗体设置：

```
@api.multi
def do_toggle_done(self):
    for task in self:
        task.is_done = not task.is_done
    return True
```

任何命令行输入或输出如下：

```
$ ~/odoo-dev/odoo/odoo-bin.py -d demo
```

新的术语和重要的词用粗体显示。例如，在屏幕上、菜单或对话框中看到的单词出现在这样的文本中：“在登录时，你会看到**应用程序**菜单，显示可用的应用程序。”



警告或重要的音符出现在这样的盒子里。



提示和技巧就像这样。

1

开始使用 Odoo开发

在进入Odoo开发之前，我们需要建立我们的开发环境，并学习它的基本管理任务。

在本章中，我们将学习如何设置工作环境，在这里我们将构建我们的Odoo应用程序。我们将学习如何设置Debian或Ubuntu系统来托管开发服务器实例，以及如何从GitHub源代码中安装Odoo。然后，我们将学习如何设置与Samba的文件共享，这将允许我们从运行Windows或任何其他操作系统的工作站运行Odoo文件。

Odoo是使用Python编程语言构建的，它使用PostgreSQL数据库进行数据存储；这些是Odoo主机的两个主要需求。要从源代码运行Odoo，我们首先需要安装它依赖的Python库。然后可以从GitHub下载Odoo源代码。虽然我们可以下载ZIP文件或tarball，但我们会看到，如果我们使用Git版本控制应用程序获取源代码会更好；它也会帮助我们把它安装在我们的Odoo主机上。

为Odoo服务器设置一个主机

一个Debian / Ubuntu系统被推荐用于Odoo服务器。你仍然可以在你最喜欢的桌面系统中工作，无论是Windows、Mac还是Linux。

Odoo可以在各种操作系统上运行，那么为什么要以牺牲其他操作系统为代价来选择Debian呢？

因为Debian被认为是Odoo团队的参考部署平台;它有最好的支持。如果我们使用Debian / Ubuntu , 它将更容易找到帮助和额外的资源。

它也是大多数开发人员工作的平台，大多数部署都是在这个平台上进行的。因此，不可避免的是，Odoo开发人员将会对Debian / Ubuntu平台感到满意。即使你是Windows背景的，你也要对它有所了解，这一点很重要。

在本章中，您将学习如何在基于debianbased的系统上设置和处理Odoo，只使用命令行。对于那些有Windows系统的家庭，我们将介绍如何设置虚拟机来托管Odoo服务器。作为一个额外的奖励，您将在这里学到的技术也将允许您在云服务器中管理Odoo，在那里您唯一的访问将通过安全Secure Shell (SSH)来访问。



请记住，这些指示是为了建立一个新的发展系统。如果您想在现有的系统中尝试其中的一些，总是提前进行备份，以便在出现问题时恢复它。

为Debian主机提供的服务

如前所述，我们需要一个基于debian-based的Odoo服务器主机。如果这是您第一次使用Linux，您可能会注意到Ubuntu是基于debianbased的Linux发行版，所以它们非常相似。

Odoo可以保证使用当前稳定版本的Debian或Ubuntu。在写作的时候，这些是Debian 8 “Jessie” 和Ubuntu 16.04.1 LTS(Xenial Xerus)。这两环境都有Python 2.7，这是运行Odoo的必要条件。值得一提的是，Odoo并不支持Python 3，因此需要Python 2。

如果你已经在运行Ubuntu或另一个基于debian-based的发行版，你就可以设置;这也可以作为Odoo的主机。

对于Windows和Mac操作系统，安装Python、PostgreSQL和所有依赖项;接下来，直接从源程序运行Odoo。然而，这可能是一个挑战，所以我们的建议是使用运行Debian或Ubuntu服务器的虚拟机。您可以选择您喜欢的虚拟化软件，以在虚拟机中获得一个工作的Debian系统。

如果您需要一些指导，这里有一些关于虚拟化软件的建议。有几个选项，比如Microsoft hyper-v(在某些版本的Windows系统中可用)、Oracle VirtualBox和VMWare工作站播放器(Mac的VMWare Fusion)。VMWare工作站的球员可能是更容易使用，并且免费下载可以在<https://my.vmware.com/web/vmware/downloads>上找到。

Regarding the Linux image to use, it will be more user-friendly to install Ubuntu Server than Debian. If you're beginning with Linux, I would recommend that you try a ready-to-use image. TurnKey Linux provides easy-to-use preinstalled images in several formats, including ISO. The ISO format will work with any virtualization software you choose, even on a bare-metal machine you might have. A good option might be the LAPP image, which includes Python and PostgreSQL, and can be found at <http://www.turnkeylinux.org/lapp>.

对于使用的Linux映像，安装Ubuntu服务器要比Debian更加友好。如果您从Linux开始，我建议您尝试使用现成的映像。TurnKey Linux提供了多种格式的易于使用的预安装映像，包括ISO。ISO格式将与您所选择的任何虚拟化软件一起工作，即使是在您可能拥有的裸金属机器上。一个很好的选择可能是第三方LAPP镜像，包括Python和PostgreSQL，可以在<http://www.turnkeylinux.org/lapp>找到。

一旦安装并启动，您应该能够登录到命令行shell。

Creating a user account for Odoo

为Odoo创建一个用户帐户

如果您正在登录使用超级用户`root`帐户，那么您的第一个任务应该是创建一个正常的用户帐户来使用您的工作，因为它被认为是不好的工作实践作为`root`。特别是，如果您将其作为`root`来启动，那么Odoo服务器将拒绝运行。

如果您正在使用Ubuntu，那么您可能不需要这个，因为安装过程必须已经通过创建一个用户来指导您。

首先，确保安装`sudo`。我们的工作用户将需要它。如果作为`root`登录，执行以下命令：

```
# apt-get update && apt-get upgrade # 安装系统更新
# apt-get install sudo # 确保安装“sudo”
```

下一组命令将创建一个`odoo`用户：

```
# useradd -m -g sudo -s /bin/bash odoo # 创建一个具有sudo能力的“odoo”用户
# passwd odoo # 请求并为新用户设置密码
```

你可以将`odoo`转换为你想要的任何用户名。 `m`选项确保创建其主目录。 `-g sudo`选项将它添加到`sudoers`列表中，以便它可以作为`root`运行命令。 `-s /bin/bash`选项将默认的shell设置为

bash，这比默认的sh要好。

现在我们可以作为新用户登录，并设置Odoo。

从源程序中安装Odoo

可以在`nightly.odoo.com`上找到现成的Odoo软件包，如Windows(.exe)、Debian(.deb)、CentOS(.rpm)和源代码tarballs(.tar.gz)。

作为开发人员，我们希望直接从GitHub存储库中安装它们。这将使我们对版本和更新有更多的控制。

为了保持整洁，在我们的主目录home内建立一个 `/odoo-dev`子目录以便进行工作。



在整本书中，我们假设 `/odoo-dev` 是您的Odoo服务器安装的目录。

首先，确保您已经登录为我们现在或在安装过程中创建的用户，而不是作为`root`用户。假设您的用户是`odoo`，请使用以下命令确认：

```
$ whoami
odoo
$ echo $HOME
/home/odoo
```

现在我们可以使用这个脚本了。它向我们展示了如何将Odoo从源代码安装到Debian / Ubuntu系统中。

首先，安装基本的依赖项，以使我们开始：

```
$ sudo apt-get update && sudo apt-get upgrade #安装系统更新
$ sudo apt-get install git # 安装Git
$ sudo apt-get install npm # 安装NodeJs及其包管理器
$ sudo ln -s /usr/bin/nodejs /usr/bin/node # 调用节点运行nodejs
$ sudo npm install -g less less-plugin-clean-css #安装less编译器
```

从版本9.0开始，Odoo web客户端需要在系统中安装less CSS预处理器，以便正确地呈现web页面。要安装这个，我们需要节点。Node.js和npm。

接下来，我们需要获得Odoo源代码并安装它的所有依赖项。Odoo源代码包括一个实用脚本，在`odoo/setup/`目录中，帮助我们在Debian / Ubuntu系统中安装所需的依赖项：

```
$ mkdir ~/odoo-dev # Create a directory to work in
$ cd ~/odoo-dev # Go into our work directory
$ git clone https://github.com/odoo/odoo.git -b 10.0 --depth=1 # Get Odoo
source code
$ ./odoo/setup/setup_dev.py setup_deps # Installs Odoo system dependencies
$ ./odoo/setup/setup_dev.py setup_pg # Installs PostgreSQL & db superuser
for unix user
```

最后，Odoo应该准备好使用。`~`符号是我们的主目录(例如，`/home/odoo`)的快捷方式。`git -b 10.0`选项告诉Git明确下载Odoo的10.0分支。在写的时候，这是多余的，因为10.0是默认的分支;然而，这可能会改变，因此它可能使脚本成为未来的证明。`--depth=1`选项告诉Git只下载最后一个版本，而不是完整的变更历史，使下载变得更小更快。

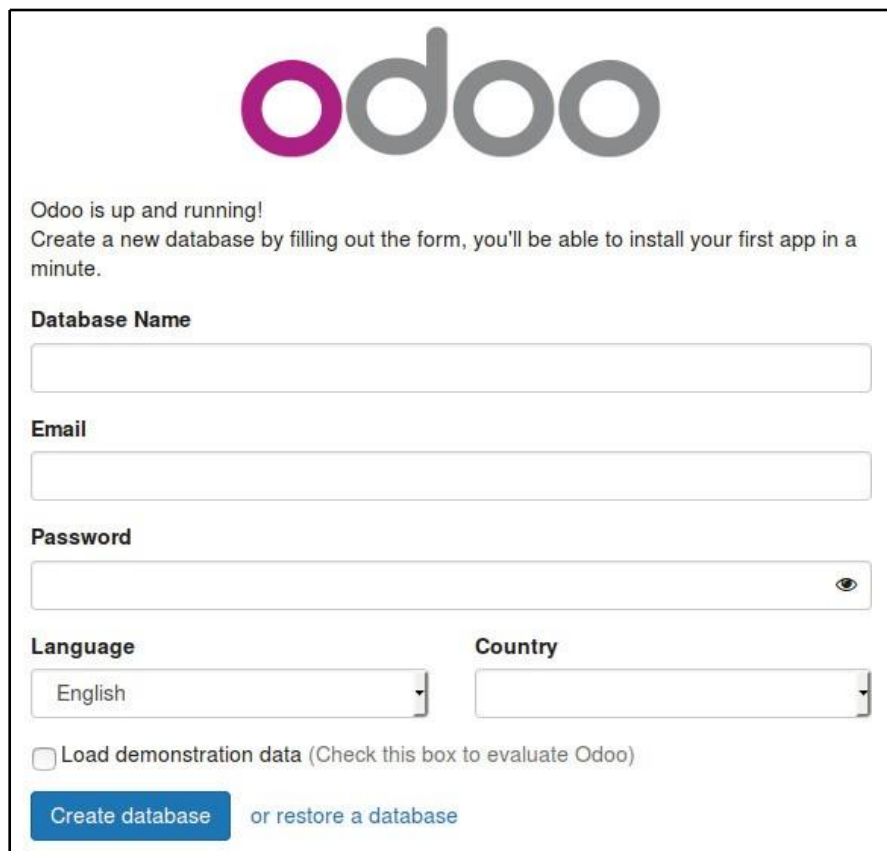
要启动一个Odoo服务器实例，只需运行：

```
$ ~/odoo-dev/odoo/odoo-bin
```



在Odoo 10中，在以前的版本中`odoo.py`脚本被`odoo-bin`替换，用于启动服务器。

在默认情况下，Odoo实例侦听端口8069，因此如果我们将浏览器指向`http://<server-address>:8069`，我们将到达这些实例。当我们第一次访问它时，它向我们展示了一个创建新数据库的助手，如下面的截图所示：



The screenshot shows the Odoo database creation wizard. At the top is the Odoo logo. Below it, a message states: "Odoo is up and running! Create a new database by filling out the form, you'll be able to install your first app in a minute." The form contains the following fields:

- Database Name**: A text input field.
- Email**: A text input field.
- Password**: A text input field with a toggle icon on the right.
- Language**: A dropdown menu currently set to "English".
- Country**: A dropdown menu.
- Load demonstration data** (Check this box to evaluate Odoo)
- Create database** (a blue button) or **restore a database** (a text link).

作为开发人员，我们需要使用几个数据库，因此从命令行创建它们更方便，因此我们将学习如何做到这一点。现在在终端按`Ctrl + C`停止Odoo服务器并返回命令提示符。

初始化一个新的Odoo数据库

为了能够创建一个新的数据库，您的用户必须是一个PostgreSQL超级用户。下面的命令为当前的Unix用户创建一个PostgreSQL超级用户：

```
$ sudo createuser --superuser $(whoami)
```

要创建一个新的数据库，请使用`createdb`命令。让我们创建一个demo数据库：

```
$ createdb demo
```

要使用Odoo数据模式初始化该数据库，我们应该使用 `-d` 选项在空数据库上运行Odoo：

```
$ ~/odoo-dev/odoo/odoo-bin -d demo
```

这将花费几分钟来初始化一个demo数据库，它将以一个信息日志消息结束，**模块加载**。

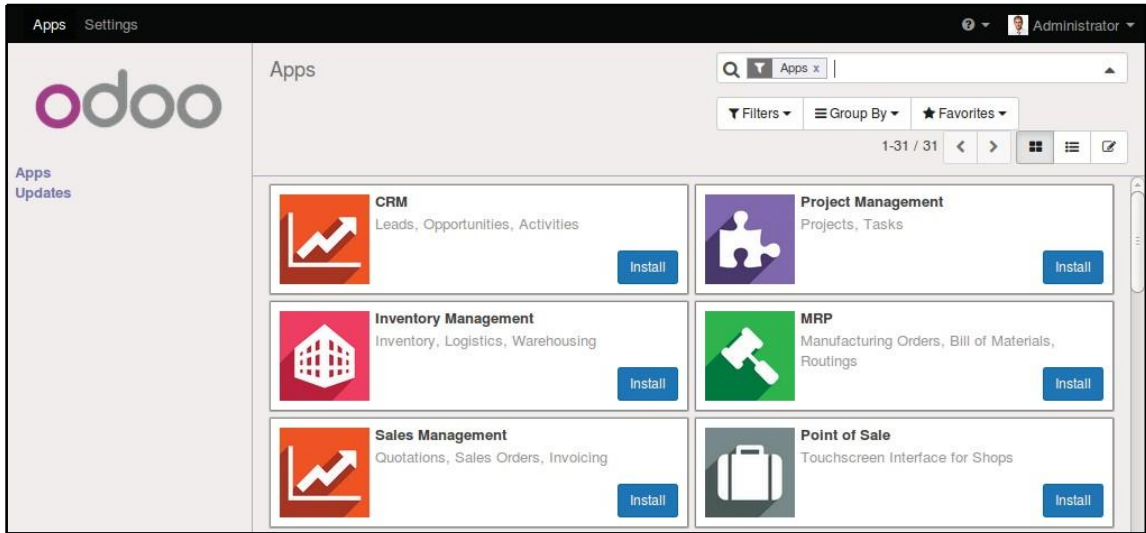


注意，它可能不是最后一个日志消息，它可以在最后三到四行。这样，服务器就可以准备好侦听客户端请求了。

默认情况下，这将用演示数据初始化数据库，这通常对开发数据库非常有用。若要初始化没有演示数据的数据库`--without-demo-data=all`。

现在打开`http://<server-name>:8069`，你的浏览器会被显示在登录屏幕上。如果您不知道您的服务器名称，在终端中键入`hostname`命令，以便找到它或`ifconfig`命令来查找IP地址。如果您在虚拟机中托管Odoo，您可能需要设置一些网络配置，以便能够从主机系统访问它。最简单的解决方案是将虚拟机网络类型从NAT改为桥接。这样，客户虚拟机就不会共享主机IP地址，而是拥有自己的IP地址。也可以使用NAT，但这需要您配置端口转发，这样您的系统就知道一些端口，比如8069，应该由虚拟机来处理。如果您遇到麻烦，希望这些细节将帮助您在您所选择的虚拟化软件的文档中找到相关信息。

默认管理员帐户是admin，其密码admin。登录后，你会看到**应用程序**菜单，显示可用的应用程序：



当您想要停止Odoo服务器实例并返回到命令行时，请在bash提示符中按`Ctrl + C`。按下向上箭头键会给你带来先前的shell命令，所以这是一个快速启动Odoo的方法，同样的选项。按下`Ctrl + C`键和向上的箭头键，进入是一个经常使用的组合，在开发期间重新启动Odoo服务器。

Managing your databases

管理数据库

我们已经了解了如何从命令行创建和初始化新的Odoo数据库。有更多的命令值得管理数据库。

我们已经知道如何使用`createdb`命令创建空的数据库，但是它也可以通过复制现有的数据库创建一个新的数据库--`template`选项

确保您的Odoo实例被停止，并且您没有打开的其他连接我们刚刚创建的演示数据库，然后运行这个：

```
$ createdb --template=demo demo-test
```

实际上，每次创建数据库时，都会使用模板。如果没有指定，则使用预定义的`template1`。

要列出系统中的现有数据库，可以使用-l选项使用PostgreSQL psql实用程序：

```
$ psql -l
```

运行它将列出我们迄今为止创建的两个数据库：demo和demo-test。该列表还将显示每个数据库中使用的编码。默认值是UTF-8，这是Odoo数据库所需的编码。

要删除不再需要的数据库(或者需要重新创建)来使用dropdb命令：

```
$ dropdb demo-test
```

现在您知道了使用数据库的基础知识。了解更多关于PostgreSQL,请参考官方文档:<http://www.postgresql.org/docs/>。



WARNING:

删除数据库命令将不可挽回地破坏您的数据。使用此命令时要小心，并且在使用此命令之前，总是要对重要的数据库进行备份。

一个关于Odoo产品版本

在写作的时候，Odoo的最新稳定版本是10版本，在GitHub上以10.0的形式标注。这是我们将在本书中使用的版本。



值得注意的是，Odoo数据库在Odoo主要版本之间不兼容。这意味着，如果您在一个以前的主要版本的Odoo创建的数据库上运行一个Odoo 10服务器，那么它就不会起作用。在使用该产品的后续版本之前，需要在数据库中使用非琐碎的迁移工作。

对于addon模块也是如此:作为一个通用规则，为Odoo主版本开发的addon模块将不会与其他版本一起工作。当从web下载一个社区模块时，确保它针对您正在使用的Odoo版本。

另一方面，主要的发行版(9.0,10.0)预计会收到频繁的更新，但这些更新大部分应该是bug修复。它们被保证为“API稳定”，意味着模型数据结构和视图元素标识符将保持稳定。这很重要，因为它意味着上游核心模块中不兼容的更改将不会导致自定义模块的破坏。

请注意，主分支中的版本将导致下一个主要的稳定版本，但在此之前，它不是“API 稳定”，您不应该使用它来构建自定义模块。这样做就像在流沙上移动：你不能确定什么时候会引入一些改变，这会破坏你的定制模块。

更多的服务器配置选项

Odoo服务器支持相当多的其他选项。我们可以查看所有可用选项`--help`：

```
$ ./odoo-bin --help
```

我们将在以下部分回顾一些最重要的选项。让我们从查看当前活动选项如何保存在配置文件中开始。

Odoo 服务器配置文件

大多数选项都可以保存在配置文件中。默认情况下，Odoo将使用`.odoorc`文件在您的主目录。在Linux系统中，它的默认位置是在`home ($HOME)`中，在Windows发行版中，它与用于启动Odoo的可执行文件处于同一目录。



在以前的Odoo / OpenERP版本中，缺省配置文件的名称为`.openerp-serverrc`。对于向后兼容性，Odoo 10仍然会使用它，如果它现在没有发现`.odoorc`文件。

在一个干净的安装上`.odoorc`配置文件不是自动创建的。我们应该使用`--save`选项创建默认配置文件，如果它还不存在，并将当前的实例配置存储到其中：

```
$ ~/odoo-dev/odoo/odoo-bin --save --stop-after-init #保存配置文件
```

在这里，我们还使用了`--stop-after-init`选项，在服务器完成其操作后停止服务器。在运行测试或要求运行模块升级以检查安装是否正确时，通常使用此选项。

现在我们可以检查在这个默认配置文件中保存的内容:

```
$ more ~/.odoorc # 显示配置文件
```

这将显示所有可使用其默认值的配置选项。在下次启动Odoo实例时，编辑它们将是有效的。键入q退出并返回到提示符。

我们还可以选择使用一个特定的配置文件，使用`--conf=<filepath>`选项。配置文件不需要您刚才看到的所有选项。只有那些真正改变了默认值的才需要在那里。

改变监听端口

`--xmlrpc-port=<port>`命令选项允许我们更改服务器实例的监听端口，从默认的8069。这可以用于在同一台机器上同时运行多个实例。

让我们试试这个。打开两个终端窗口。首先，运行这个:

```
$ ~/odoo-dev/odoo/odoo-bin --xmlrpc-port=8070
```

在第二个终端上运行以下命令:

```
$ ~/odoo-dev/odoo/odoo-bin --xmlrpc-port=8071
```

在这里，两个Odoo实例在同一个服务器上监听不同的端口!这两个实例可以使用相同或不同的数据库，这取决于所使用的配置参数。这两个版本可以运行相同或不同版本的Odoo。

数据库过滤选项

与Odoo一起开发时，它经常与几个数据库一起工作，有时甚至使用不同的Odoo版本。在同一个端口上停止和启动不同的服务器实例，并在不同的数据库之间切换，会导致web客户端会话的行为不正确。

使用在私有模式下运行的浏览器窗口访问我们的实例可以帮助避免这些问题。

另一个好的做法是在服务器实例上启用数据库过滤器, 以确保它只允许我们想要处理的数据库的请求, 而忽略所有其他的。这是用`--db-filter`选项完成的。它接受一个正则表达式作为有效数据库名称的过滤器。匹配一个确切的名字, 表达应该开始`^`和结束`$`。

例如, 为了只允许演示数据库, 我们将使用这个命令:

```
$ ~/odoo-dev/odoo/odoo-bin --db-filter=^demo$
```

管理服务器日志消息

`--log-level` 选项允许我们设置log verbosity。这对于了解服务器上正在发生的事情非常有用。例如, 要启用调试日志级别, 请使用`--log-level=debug` 选项。

以下日志级别特别有趣:

`debug_sql` 检查服务器生成的SQL查询

`debug_rpc` 详细说明服务器接收到的请求

`debug_rpc_answer` 详细说明服务器发送的响应

默认情况下, 日志输出被定向到标准输出(您的控制台屏幕), 但是它可以直接指向一个日志文件, 其中包含`--logfile=<filepath>`选项。

最后, 当一个异常被提起时, `--dev=all`选项将打开Python调试器(pdb)。对服务器错误进行事后分析是很有用的。注意, 它对logger verbosity没有任何影响。更多细节在Python调试器命令可以在<https://docs.python.org/2/library/pdb.html>找到debugger-commands。

从您的工作站

您可能正在使用一个Debian / Ubuntu系统在本地虚拟机或网络服务器上运行Odoo。但您

可能更喜欢在您的个人工作站使用您喜欢的文本编辑器或IDE进行开发工作。对于Windows工作站的开发人员来说，这可能是常见的情况。但对于那些需要在本地网络上使用Odoo服务器的Linux用户来说，情况也是如此。

解决这个问题的方法是在Odoo主机中启用文件共享，这样就可以方便地从我们的工作站编辑文件。对于Odoo服务器操作，比如服务器重启，我们可以在我们最喜欢的编辑器旁边使用SSH shell(比如在Windows上的PuTTY)。

使用Linux文本编辑器

迟早，我们需要从shell命令行编辑文件。在许多Debian系统中，默认的文本编辑器是vi。如果您对它不满意，您可能可以使用更友好的选择。在Ubuntu系统中，默认的文本编辑器是nano。你可能更喜欢它，因为它更容易使用。如果您的服务器上没有可用的，可以安装：

```
$ sudo apt-get install nano
```

在下面的章节中，我们将假设nano是首选的编辑器。如果您喜欢其他的编辑器，请随意调整相应的命令。

安装和配置Samba

Samba服务帮助使Linux文件共享服务与Microsoft Windows系统兼容。我们可以在Debian / Ubuntu服务器上安装这个命令：

```
$ sudo apt-get install samba samba-common-bin
```

samba包安装了文件共享服务，smbpasswd工具需要使用samba-common-bin包。默认情况下，允许访问共享文件的用户需要注册。我们需要注册我们的用户odoo，并为其文件共享访问设置密码：

```
$ sudo smbpasswd -a odoo
```

在此之后，我们将被要求使用一个密码来访问共享目录，而odoo用户将能够访问其主目录的共享文件，尽管它只会被读取。我们想要有写访问权限，因此我们需要编辑Samba配置文件，以更改以下内容：

```
$ sudo nano /etc/samba/smb.conf
```

在配置文件中，查找[homes]部分。编辑其配置行，使其与以下设置匹配：

```
[homes]
  comment = Home Directories
  browseable = yes
  read only = no
  create mask = 0640
  directory mask = 0750
```

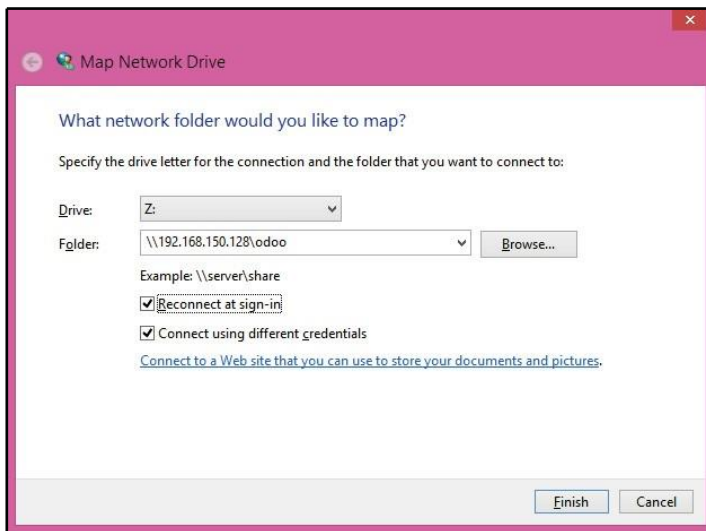
对于配置更改生效，重新启动服务：

```
$ sudo /etc/init.d/smbd restart
```

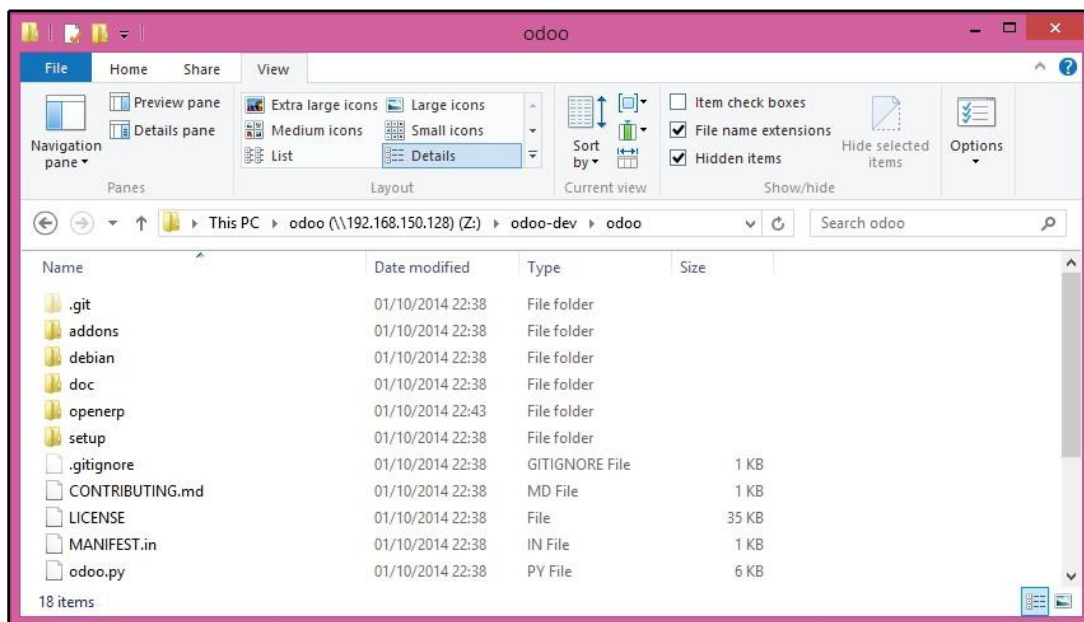
Downloading the example code



要从Windows中访问这些文件，我们可以使用与smbpasswd定义的特定的用户名和密码，将网络驱动器映射为\\<my- server-name>\odoo。当试图与odoo用户登录时，您可能会遇到Windows向用户名(例如MYPC\odoo)添加计算机域的问题。为了避免这种情况，请在登录时使用一个空域(例如，\odoo)：



如果我们现在使用Windows资源管理器打开映射驱动器，我们将能够访问和编辑odoo用户的主目录的内容：

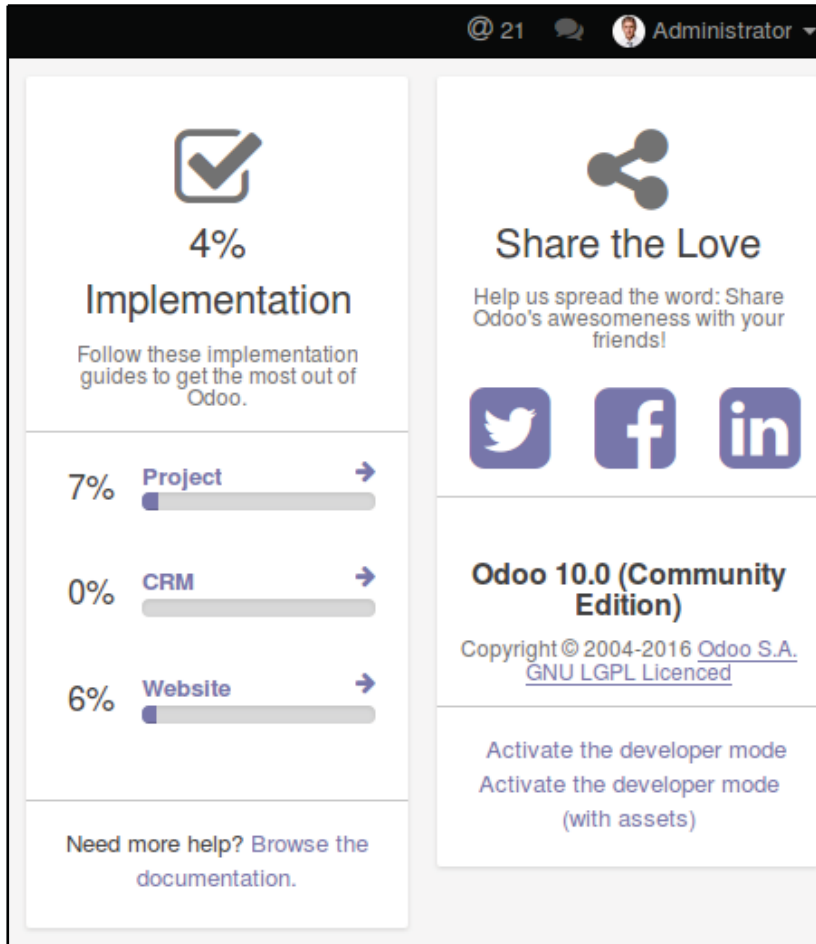


Odoo包括一些对开发人员非常有用的工具，我们将在本书中使用它们。它们是技术特性和开发模式。默认情况下，这些都是禁用的，所以这是一个学习如何启用它们的好时机。

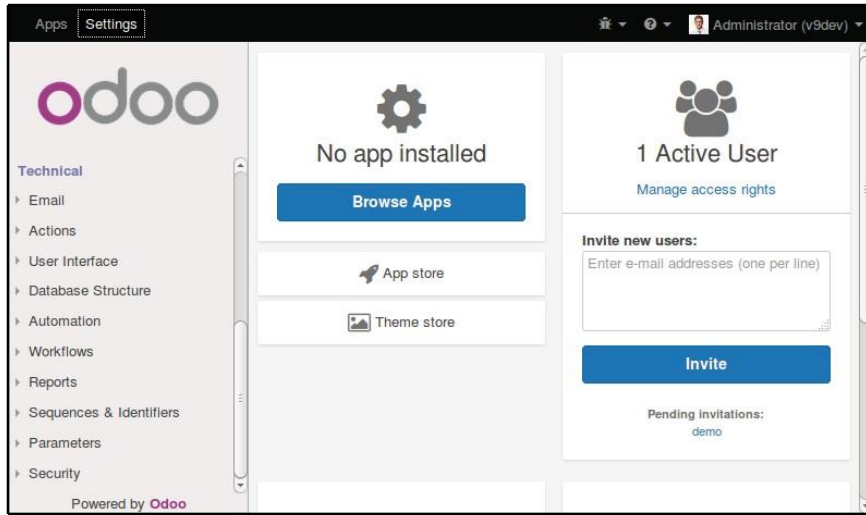
激活开发工具

开发工具提供高级服务器配置和特性。其中包括顶部菜单栏中的一个调试菜单，以及**设置菜单**中的其他菜单选项，特别是**技术菜单**。

这些工具在默认情况下是禁用的，并且为了启用它们，我们需要以admin身份登录。在顶部菜单栏中，选择**Settings**菜单。在底部右侧，在Odoo版本之下，您将找到两个选项来启用开发者模式；它们中的任何一个都可以**启用调试和技术菜单**。第二个选项，**激活开发者模式(with assets)**，也禁用了web客户端使用的JavaScript和CSS，这样可以更容易地调试客户端行为：



在此之后，页面被重新加载，您应该在顶部菜单栏中看到一个bug图标，就在会话用户头像和提供调试模式选项的名称之前。在顶部菜单的**Settings**选项中，我们应该看到一个新的**技术菜单**部分，可以访问许多Odoo实例内部：



技术菜单选项允许我们检查和编辑存储在数据库中的所有Odoo配置,从用户界面到安全性和其他系统参数。在这本书中,你将会学到更多。

安装第三方模块

在一个Odoo实例中提供新的模块,这样它们就可以安装,这是一个新手经常会感到困惑的东西。但不一定非得如此,让我们来揭开它的神秘面纱吧。

Finding community modules

互联网上有很多的Odoo模块。odoo.com是一个可以在你的系统上下载和安装的模块目录。**Odoo社区协会(OCA)**协调社区贡献,并在<https://github.com/OCA/>上维护了GitHub上的一些模块存储库。

要在Odoo安装中添加一个模块，我们可以将它复制到addons目录中，并与官方模块一起使用。在我们的案例中，addons目录位于~/odoo-dev/odoo/addons/。对于我们来说，这可能不是最好的选择，因为我们的Odoo安装是基于版本控制的代码存储库，我们将希望它与GitHub存储库保持同步。

幸运的是，我们可以为模块使用额外的位置，这样我们就可以将自定义模块保存在不同的目录中，而不需要将它们与官方的模块混合在一起。

作为一个例子，我们将从这本书中下载代码，在GitHub中提供，并使那些addon模块在我们的Odoo安装中可用。

要从GitHub获得源代码，运行以下命令：

```
$ cd ~/odoo-dev
$ git clone https://github.com/dreispt/todo_app.git -b 10.0
```

我们使用 `-b` 选项确保我们正在下载10.0版本的模块。

在此之后，我们将在/odoo目录旁边有一个新的/todo_app目录，其中包含模块。现在我们需要让Odoo知道这个新的模块目录。

配置插件的路径

Odoo服务器有一个名为`addons_path`的配置选项，用于设置服务器应该在哪里查找模块。默认情况下，这指向/odoo/addons目录，在那里，Odoo服务器正在运行。

我们不仅可以提供一个目录，还可以提供一个目录列表，其中可以找到模块。这允许我们将自定义模块保存在一个不同的目录中，而不让它们与正式的addons混合。

让我们以一个包含我们的新模块目录的addons路径启动服务器：

```
$ cd ~/odoo-dev/odoo
$ ./odoo-bin -d demo --addons-path="../todo_app,./addons"
```

如果您仔细查看服务器日志，您将注意到一条正在使用的addons路径的行：`INFO? odoo: addons paths: [...].` 确认它包含我们的todo_app目录。

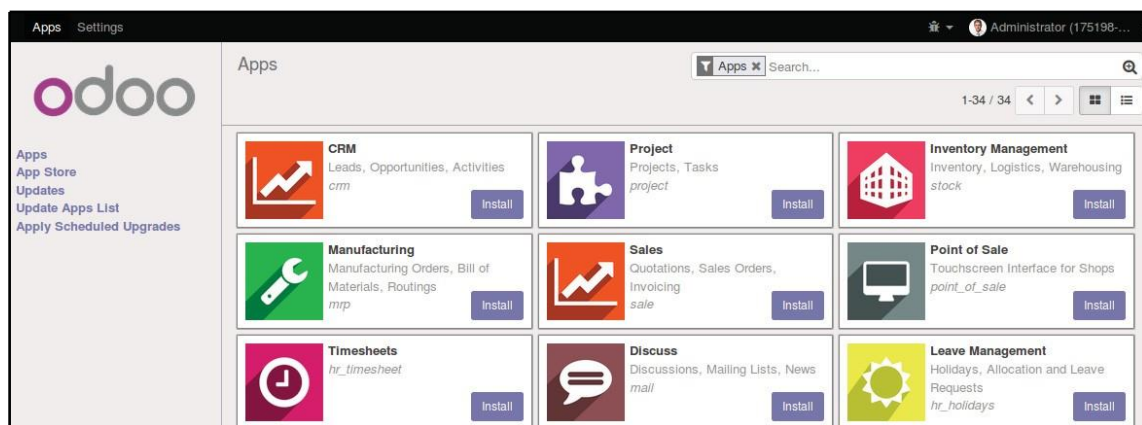
更新应用程序列表

我们还需要让Odoo在这些新模块提供安装之前更新它的模块列表。

为此，我们需要启用developer模式，因为它提供了**Update Apps**列表菜单选项。它可以在**应用程序**头部菜单中找到。

更新模块列表后，我们可以确认新的模块可供安装。使用**应用程序**菜单选项查看本地模块列表。搜索todo，您应该看到可以使用的新模块。

注意，第二个**App Store**菜单选项显示的是Odoo应用商店的模块列表，而不是本地模块：



摘要

在本章中，我们学习了如何设置一个Debian系统来托管Odoo并将其安装到GitHub源代码中。我们还学习了如何创建Odoo数据库并运行Odoo实例。为了允许开发人员在其个人工作站上使用他们最喜欢的工具，我们解释了如何在Odoo主机中配置文件共享。

我们现在应该有一个运行良好的Odoo环境，并对管理数据库和实例感到满意。

有了这个，我们就可以直接开始行动了。在下一章中，我们将从头开始创建我们的第一个Odoo模块，并理解它所涉及的主要元素。

所以让我们开始吧!

