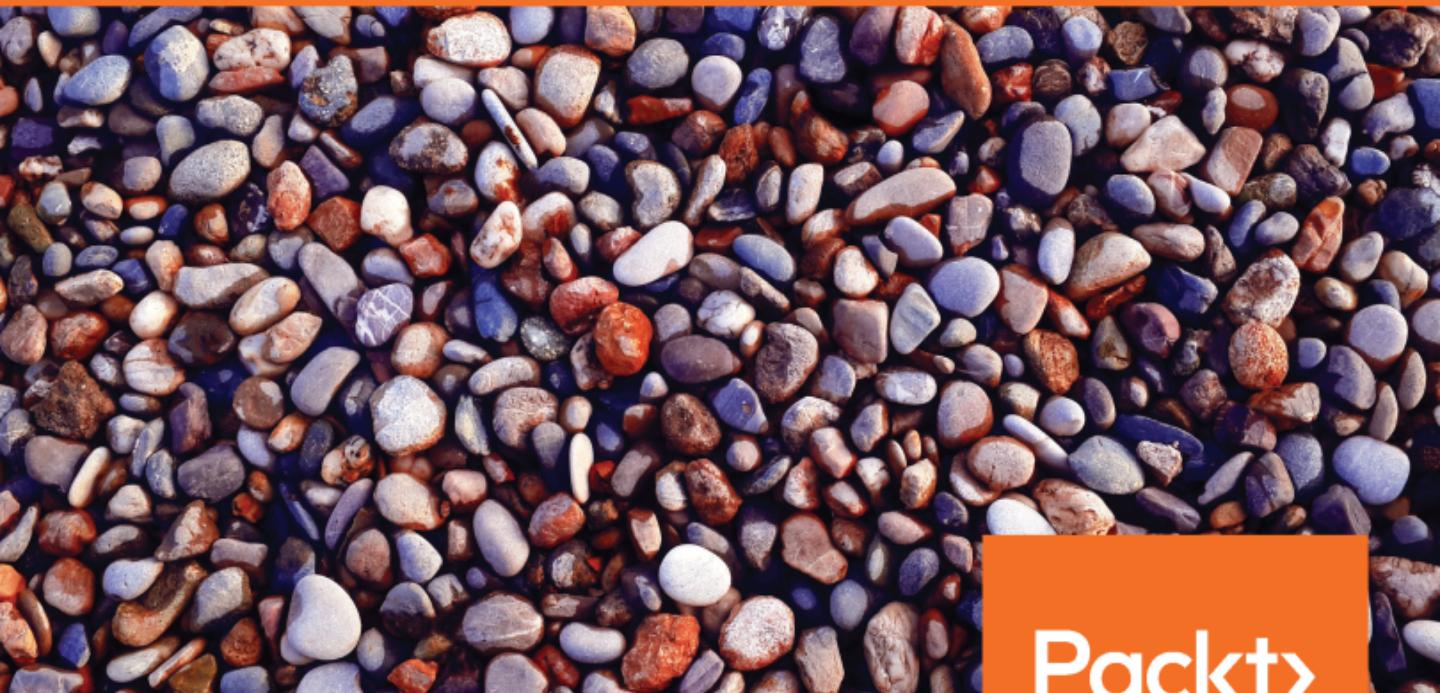


Learn Odoo

A beginner's guide to designing, configuring, and customizing business applications with Odoo



Greg Moss

Packt

www.packt.com

Learn Odoo

A beginner's guide to designing, configuring, and customizing business applications with Odoo

Greg Moss

Packt

BIRMINGHAM - MUMBAI

Learn Odoo

Copyright © 2019 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavoured to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Pavan Ramchandani

Acquisition Editor: Denim Pinto

Content Development Editor: Akshita Billava

Senior Editor: Afshaan Khan

Technical Editor: Ashi Singh

Copy Editor: Safis Editing

Project Coordinator: Prajakta Naik

Proofreader: Safis Editing

Indexer: Tejal Daruwale Soni

Production Designer: Alishon Mendonsa

First published: October 2019

Production reference: 1311019

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-78953-689-8

www.packt.com



Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Contributors

About the author

Greg Moss has been a business and information systems consultant for over 30 years and is both a **Certified Information Systems Auditor (CISA)** and a Certified Six Sigma Black Belt. He also has an Information Assurance Certification from Carnegie Melon University. Presently, Greg is the CEO of First Class Ventures, LLC., the owner of First Class Computer Consulting—Odoo consulting and training, the Odoo Class website, and an Odoo Ready partner.

Greg is the founder of Odoo Class, a technical training platform for Odoo. He has published more than 20 courses on Udemy.

In his spare time, Greg enjoys playing the trumpet, saxophone, keyboard, and guitar, and taking cross-country road trips with his African grey parrot, Bibi.

About the reviewer

Bruce Letterle is a veteran of the IT and web industries and has worked in this field since 1979. He is the founder and managing director of Red Lab Media, a provider of web development, marketing services, hosting, and Odoo implementation services based in Pittsburgh, PA, USA. His company is an Odoo partner and has been active with Odoo products since 2010 when they were known as OpenERP.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit authors.packtpub.com and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Preface	1
Chapter 1: Setting Up Odoo 12	8
Community and Enterprise Editions of Odoo	9
Getting started with Odoo Online	9
Taking advantage of Odoo Online instant access	10
Odoo browser requirements	10
Odoo mobile phone and tablet support	11
Accessing the Odoo free online trial	11
Continuing to use the trial version of Odoo	15
Subscribing to Odoo	15
Using Odoo without subscription fees	17
Getting to know the Odoo architecture	17
Introducing the PostgreSQL database	17
Writing code with Python	18
Following the Model-View-Controller design	18
Designing models	18
Rendering views	18
Authoring controllers	19
Choosing your installation OS	19
Choosing a Microsoft Windows Odoo installation	19
Learning Ubuntu is not required	19
Introducing Ubuntu	19
Choosing an Ubuntu Odoo installation	20
Choosing another OS option for Odoo	20
Understanding Odoo releases	21
Upgrading Odoo	21
Installing Odoo on Windows OS	21
Performing an all-in-one Odoo installation on Windows	23
Configuring Postgres on Windows	23
Installing Odoo on Ubuntu	25
Modifying the sources.list file	25
Testing your Odoo installation	26
Troubleshooting and Odoo management tips	26
Checking your browser destination	26
Verifying that the Odoo service is running	26
Checking for the Odoo services running in Windows	27
Checking for Odoo services running in Ubuntu	28
Starting and stopping Odoo services in Ubuntu	28
Finding the primary Odoo log file	29
Modifying the Odoo configuration file	29

Changing port numbers	29
Accessing the database management tools	29
Changing the admin password	30
Finding additional resources on installing Odoo	30
Summary	30
Chapter 2: Installing Your First Application	31
Gathering requirements	32
Implementing Odoo using a modular approach	32
Introducing Silkworm – our real-world case study	32
Creating a new database in Odoo	33
Selecting a database name	34
Choosing an email address and password	34
Specifying a phone number	35
Specifying our default language	35
Specifying the country	35
Loading demonstration data	35
Managing databases in Odoo	36
Installing the Sales application	36
Knowing the basic Odoo interface	37
Configuring company data	38
Selecting your quotation layout	40
Specifying payment options	40
Sending a sample quotation	41
Creating your first customer	42
Specifying whether the customer is an individual or a company	43
Entering data into an Odoo form	44
Using multiple contacts and addresses	45
Editing customer Sales & Purchases	45
Editing customer accounting	47
Entering a product in Odoo	47
Creating products in Odoo	48
Product name	48
Can be sold	48
Can be purchased	49
Product type	49
Internal reference	49
Sale price	49
The Cost price	49
Checking out the Sales tab	50
Entering a product Invoicing page	50
Invoicing policy	51
Saving the product	51
Entering your first sales order	52
Selecting the customer	53
The Validity date	54
Payment terms	54

Entering line items on a quotation sales order	55
The Product field	55
Description	56
Ordered quantity	56
Taxes	56
Unit price	56
Saving the quotation sales order	57
Understanding the sales order workflow	57
Send by Email	58
Print	58
Confirm Sale	59
Cancel	59
Invoicing the sale	60
What do you want to invoice?	60
Creating the invoice	61
Summary	62
Chapter 3: Exploring Customer Relationship Management in Odoo 12	63
Using CRM as a business strategy	64
Managing customer experience	64
Treating your customer like a real person	64
Using your mission statement to drive the design of your CRM system	65
A real-world case study – improving customer experience	65
Installing the CRM application	66
Your first look at the Odoo CRM pipeline	67
Assigning the sales representative or account manager	67
Specifying the name of the user	68
Email address	69
Access Rights	69
Sales application settings	69
User – own documents only	69
User – all documents	70
Manager	70
Language and timezone	71
Messaging and social – notification management	71
Signature	71
Manually setting the user password	71
Assigning a salesperson to a customer	72
Understanding your pipeline	73
Creating a new opportunity	74
Opportunity	75
Customer	75
Expected revenue	75
Rating	76
Looking at opportunities in your Pipeline	76
Expected closing	78
Tags	78
Email and Phone	78
Internal Notes	78
Contact information	78

Address and contact information	79
Mailings	79
Marketing	80
An introduction to sales stages	80
The Activities menu	82
Viewing the details of an opportunity	82
Using Odoo's Chatter feature	83
Changing the status of an opportunity	83
Managing the opportunity	85
Following a business document	86
Modifying the stages of the sale	87
Closing the sale	89
Your opportunity converted to a quotation	90
Leads and opportunities	90
Turning on leads in Odoo 12	90
Creating leads in Odoo	91
Converting a lead into an opportunity	93
Using Odoo to schedule calls, meetings, and events	94
Scheduling an event	95
Adding attendees to your meeting	96
Specifying meeting details	97
Specifying tags for your meeting	97
Setting up reminders for your meeting	98
Specifying additional meeting options	98
Summary	99
Chapter 4: Purchasing with Odoo	100
Understanding the overall purchasing process	100
Setting up a vendor	101
Setting up warehouse locations	101
Generating quotations and purchase orders	101
Receiving the product	102
Settling the invoice	102
Installing the Purchase application	102
Setting up your first vendor	103
Designating a vendor as Individual or Company	104
Product type	105
Setting the cost price of the product	105
Assigning vendors to the product	105
Establishing the vendor	106
Designating the vendor product name and product code	107
Setting minimal quantity	107
Calculating delivery time	107
Setting price and validity	108
Creating your first purchase quotation	108
Adding products to your request for quotation	109
Printing RFQs and updating status	110

Confirming a purchase order	110
Receiving products	111
Creating a vendor bill	111
Paying vendor bills without the purchase order	112
Handling complex units of measure	116
Looking at the defined units of measure	117
Examining a specific Unit of Measure	118
Configuring a product to be purchased in dozens	119
Summary	120
Chapter 5: Making Goods with Manufacturing	121
Creating manufacturing orders	122
Producing the product	122
Delivering the order	122
Defining the workflow for your business	122
A real-world example of a custom product	123
Installing Manufacturing	123
Creating your first manufacturing order	124
What product are we going to manufacture?	125
Building your Bill of Materials	127
Setting a product as storable	130
Confirming production	131
Checking availability	132
Producing the product	133
Analyzing stock valuation	135
Managing production by work orders	135
Creating a work center	137
Setting general information	138
Time Efficiency	139
Capacity	139
Overall equipment effectiveness target	139
Time before and after production	139
Creating routing orders	140
Assigning the routing to a BoM	142
Creating a manufacturing order with routing and a work center	143
Creating routings based on product variants	147
Controlling the creation of variants	149
Adding a size attribute	149
Using unbuild orders or a reverse BoM	151
Creating a kit using a BoM	152
Summary	153
Chapter 6: Configuring Accounting - Finance Options	154
Defining the chart of accounts for your business	155
Configuring a user to see all Community Edition accounting options	155

Using the configuration panel for accounting	159
Reviewing the current chart of accounts	162
Getting more information on a specific account	163
Completing the accounting configuration	165
Learning how transactions in Odoo get posted to accounts	165
Following transactions through the sales and accounts receivable process	
Viewing the transactions that have been created by validating the invoice	169
Practicing posting transactions and tracking the results	173
Setting up your own accounts	
Specifying a new account for your product category	177
Practicing with another product	179
Examining the available legal reports in Odoo	181
Creating journal entries	181
Summary	182
Chapter 7: Administering an Odoo Installation	184
Basic considerations for an Odoo administration	185
Have an implementation strategy	186
Development, staging, and production servers	186
Clear documentation of all Odoo configurations	187
Focusing on business continuity	187
Backing up your Odoo database	188
Restoring an Odoo database	190
Administering users in Odoo	191
Selecting a user to administer	191
Managing user preferences	193
Understanding groups in Odoo	195
Understanding group inheritance in Odoo	197
Defining menus for your group	198
Understanding access rights in Odoo	199
Understanding group rules in Odoo	202
Internationalization in Odoo	204
Configuring language translation	205
International currencies	208
Purchasing in a different currency	212
Managing supplier price lists	213
Managing sequences in Odoo	217
Multiple companies in Odoo	221
Setting up a second company in Odoo	221
Setting up the chart of accounts for your second company	225
Implementing a multi-company solution	228
Summary	228
Chapter 8: Implementing the Human Resources Application	229
Modular approach to HR applications	229
Installing the Employee Directory	230

Creating a new employee	231
Department	231
Job Title	234
Manager and Coach	235
Employee Private Information	236
HR Settings	237
Managing departments	237
Timesheets	238
Leave management	241
Creating a leave request	242
Leave type	242
Duration	243
Mode	243
Employee	243
Department	243
Submitting for approval	243
Approving leave requests	245
Recruitment process	245
Recruiting for a new job position	246
Creating an employment application	248
Hiring employees	250
Summary	251
Chapter 9: Understanding Project Management	252
What's new in Odoo 12?	252
The basics of Project Management	253
Installing the Project Management application	253
Understanding the Project dashboard	254
A real-world project example for a customer	254
Creating our first project	255
Understanding project stages	256
Defining project tasks	258
Creating additional tasks	260
Completing project tasks	261
Scheduling an activity	262
Monitoring task completion	263
Calculating project costs and time	264
Summary	265
Chapter 10: Creating Advanced Searches and Dashboards	266
Determining the search requirements for your business	266
Creating a database with demonstration data	267
Accessing the database manager	267
New demonstration data in Odoo 12	270
Searching in Odoo	271

Using filters in the list views	275
Grouping information	277
Performing a custom search	279
Specifying multiple advanced searches	282
Saving your favorite filters	283
Creating custom dashboards in Odoo 12	285
Looking at Odoo's business intelligence features	290
Summary	293
Chapter 11: Building a Website with Odoo	294
What is a CMS?	294
WordPress	295
Joomla	295
Drupal	295
Evoq or DotNetNuke	295
Why use Odoo Website Builder as your CMS?	296
Potential advantages of using Odoo as a CMS	296
Current limitations of using Odoo as a CMS	297
Backing up the website you make in your Odoo database	297
Installing the Odoo Website Builder	298
Saving your web page	307
Previewing our website on a mobile device	309
Adding new pages and menus to your website	310
Adding a new page	311
Creating our frequently asked questions web page	313
Managing menus on your website	314
Adding a new menu	314
Changing themes in Odoo	315
Promoting your website	317
Specifying keywords for your website	319
Creating a title and description for your website	320
Summary	321
Chapter 12: Implementing E-Commerce with Odoo	322
eCommerce and Odoo	322
Popular eCommerce platforms	323
Magento	323
Volusion	323
Shopify	323
Yahoo Small Business	324
Odoo as an eCommerce platform	324
Installing Odoo eCommerce	324
Basics of Odoo eCommerce	326
Setting the product price from the eCommerce page	328
Adding a picture to your product	329
Describing the product	331

Publishing your product	332
Additional configuration options for your Odoo shop	333
Modifying the order of the products on the store	335
Setting up product variants	337
Advanced eCommerce product options	340
Alternative products	342
Accessory products	342
Looking at the shopping cart	342
Seeing the draft sales order in Odoo	344
Checking out	345
Adding PayPal as a payment processor	348
Summary	350
Chapter 13: Customizing Odoo for Your Business	351
 Understanding the Odoo architecture	352
Models	352
Views	352
Actions	352
 Activating developer mode	353
Getting out of developer mode	355
 Backing up your database	355
Restoring a database in Odoo	356
 Adding a custom field to Odoo	357
Viewing the model in Odoo	358
Creating a new field in the sale.order model	360
Editing the form view	362
Customizing search operations in Odoo	366
Specifying additional fields	366
Editing the search view	368
 Understanding actions	370
 Creating a new menu	372
 Integrating applications using Odoo's API	374
 Summary	375
Chapter 14: Modifying Documents and Reports	376
 Developing the skills that are required to modify reports	377
 What is the QWeb template engine?	377
 Company report configuration	378
Problems with the Wkhtmltopdf installation	381
Specifying the paper format for your reports	381
 Making our first simple change	383
Modifying the internal report header using QWeb	385
 Learning how Odoo organizes reports	387
 Understanding the report types	388
 Looking at the definition for the Sales order form	388

Creating a new QWeb report in Odoo	393
Learning more about the power of QWeb templates	393
Using a t-if to create a dynamic QWeb template report	394
Summary	394
Chapter 15: Discovering Custom Odoo Modules	395
Exploring the Odoo application and module directory	396
Building our first Odoo module	396
Preparing your basic development environment	397
Finding your Odoo configuration file and installing it	397
Specifying a custom directory to hold our Odoo module	398
Contents of your module directory	400
Creating and editing the files	400
The <code>__init__.py</code> file	400
The <code>__manifest__.py</code> file	401
Extending an Odoo model in <code>silkworm.py</code>	403
Using a module to add custom fields to a model	403
Inheriting from the Odoo Sales application	404
Understanding Python conventions	405
Adding the fields to our sales order view	405
Getting ready to install our module	408
Troubleshooting your module installation	410
Using a module to add a filter to a search view	412
Adding the code to create the rush order filter	413
Creating the filter	414
Using Odoo to create websites and web services	414
Creating an XML template	416
Integrating with the Odoo API	417
Connecting to the API	418
Filtering and returning records through the API	418
Using the <code>search_read</code> method	418
Creating custom themes in Odoo	419
Basic architecture of Odoo themes	419
Creating an Odoo theme using scaffolding	419
Modifying the default Odoo theme header	420
Creating a custom page layout	422
Summary	424
Chapter 16: Comparative Analysis of Community versus Enterprise Editions	425
Getting an overview of the Community and Enterprise Editions	426
Using the Odoo runbot to compare Odoo versions	427
Launching the Odoo Community Edition from Odoo runbot	428
Launching Odoo 12 Enterprise from Odoo runbot	429
Examining the Odoo Enterprise interface	430
Examining Odoo 12 Enterprise Accounting	431
Looking at Odoo 12's dynamic accounting reports	432

Finding out which options are only available for Enterprise	436
Sales-related enhancements	437
Automating calls with VoIP integration	437
Understanding the Subscriptions application	438
Understanding the available shipping connectors in Enterprise	440
Understanding the available manufacturing applications in Odoo	442
Product Lifecycle Management (PLM)	443
Handling maintenance requests	445
Managing production quality in Odoo	447
Using Odoo Studio	450
Modifying the Sales application with Odoo Studio	450
Summary	453
Appendix A: Locating Additional Odoo Resources	454
Locating the essential Odoo documentation	454
Visiting the official Odoo help forum	454
The Odoo Community Association	454
Finding Odoo applications and modules	455
Getting the latest Odoo 12 release notes	455
Downloading Odoo from GitHub	455
Locating resources on Ubuntu	455
Official Ubuntu website	455
Directly downloading the Ubuntu server or desktop	456
Official Ubuntu documentation	456
Getting additional developer documentation	456
Getting quick access to Odoo installations using Odoo Runbot	456
Finding Postgres resources	457
Official Postgres website	457
Downloading Postgres	457
Postgres documentation	457
Locating Python resources	457
Official Python website	457
Downloading Python	458
Python documentation	458
Finding XML resources	458
World Wide Web Consortium XML resource page	458
XML tutorials and documentation	458
Other Books You May Enjoy	459
Index	462

Preface

Learn Odoo provides a comprehensive walk-through of installing, configuring, and implementing Odoo in real-world business environments. This book will assist you in understanding the value of Enterprise Resource Planning (ERP) systems and best-practice approaches for getting a system up and running in your organization. For those that are new to ERP systems, this book will serve as an introduction so that you will be better prepared to understand more advanced ERP concepts. If you are already experienced in ERP systems, this book will give you an overview of the primary applications for Odoo and how those applications can be used in a real business environment.

Odoo is a very feature-filled business application framework, with literally hundreds of applications and modules available. In fact, in recent years, Odoo has often not even been referred to as an ERP, because it is capable of solving such a variety of solutions. We have done our best in this book to cover the most essential features of the Odoo applications you are most likely to use in your business. Unfortunately, there are just not enough pages to cover more advanced topics. In the [Appendix A, Locating Additional Odoo Resources](#), we have included additional resources that you can look to for more advanced subjects.

Once the book has been published, be sure to check for online updates in which we'll cover more advanced subjects.

Who this book is for

This book is for everyone who is interested in implementing an ERP system in a business organization. If you are an IT professional looking to get a functional understanding of Odoo, then this book is for you. This book is also appropriate for business and operations managers who wish to get a comprehensive understanding of Odoo and how it can be used to improve business processes.

What this book covers

The book is divided into the following sections:

1. Installation of Odoo and the basics for implementing Odoo in your business (Chapter 1, *Setting Up Odoo 12*, to Chapter 5, *Making Goods with Manufacturing*)
2. Introduction to accounting and finance setup and modules to help your business run more efficiently (Chapter 6, *Configuring Accounting - Finance Options* to Chapter 8, *Implementing the Human Resources Application*)
3. Advanced configurations and customization of Odoo (Chapter 9, *Understanding Project Management* to Chapter 15, *Discovering Custom Odoo Modules*)
4. A comprehensive comparison of Odoo Community and Odoo Enterprise (Chapter 16, *Comparative Analysis of Community versus Enterprise Editions*)

Chapter 1, *Setting Up Odoo 12*, gets you started right away by showing how you can use Odoo online without any setup. Just open up your browser and you are ready to get going. Next, the chapter goes on to cover the different installation types and prerequisites for both Windows and Ubuntu. Instructions are provided on finding the right download package and setting up Odoo on your own server. The chapter then goes into the basics of configuring Odoo. At the end of the chapter is a useful collection of tips on how to troubleshoot your Odoo installation.

Chapter 2, *Installing Your First Application*, begins by introducing you to the real-world case study that will be used as an example throughout the book. We continue by learning how to create the company database and configure the basic company settings required to quickly get your first Odoo system up and running. The first module, Sales Management, will be installed, and we will walk through the steps for entering a customer and a product. The chapter concludes with entering a sales order, completing the sale, and producing an invoice.

Chapter 3, *Exploring Customer Relationship Management in Odoo 12*, starts with a basic overview of CRM systems and their importance in today's modern business environment. After we cover the installation of the CRM module, a lead is entered for our sample company. We will demonstrate the CRM workflow by turning the lead into a customer. Next, a quote is generated for our newly acquired customer, and a call is scheduled for follow-up using Odoo's meeting functionality. We also cover the Open Chatter feature, which is used throughout Odoo to provide notes and messages associated with Odoo documents.

Chapter 4, *Purchasing with Odoo*, shows you how to install the purchasing module, set up suppliers, and begin purchasing and receiving products through Odoo. Later in the chapter, you'll learn how to tie purchasing into sales orders to automatically generate draft purchase orders based on your business requirements.

Chapter 5, *Making Goods with Manufacturing*, begins to explore some of the primary functionality of ERP systems for manufacturing operations. You will learn how to set up your manufacturing orders and define the bill of materials to specify the raw materials that will go into your final products. Manufacturing operations can then be extended with routing and work centers to provide you with more control when tracking time and resources.

Chapter 6, *Configuring Accounting - Finance Options*, discusses the Accounts Receivable and Accounts Payable basic functions. Next, we introduce the chart of accounts and discover how to set up fiscal periods. This chapter also covers basic accounting reports and how to close a period.

Chapter 7, *Administering an Odoo Installation*, begins by discussing overall considerations for implementing Odoo in a business environment. This includes advice on server configurations, documenting your processes, and the importance of considering business continuity. We then go into how to manage users, groups, and set up security roles to manage access to various applications within Odoo. Finally, we look at how to implement internationalization for multiple languages and currencies.

Chapter 8, *Implementing the Human Resources Application*, begins by installing the basic Human Resources (HR) modules and going over the employee directory. Other topics in the chapter include timesheets, the recruitment process, and leave management. At the end of the chapter, we look at how to create online interviews and hire employees using the tools in Odoo.

Chapter 9, *Understanding Project Management*, covers the features of the Project Management module in Odoo. We will create a project, and see how to enter tasks and tie a project to a specific customer. Next, team members are assigned to the project, and we configure task stages. We then go over real-world examples of using the Project Management module to more easily manage complex orders and customer needs. Finally, we see how Project Management can be used along with analytic accounting to provide better reporting.

Chapter 10, *Creating Advanced Searches and Dashboards*, demonstrates how to utilize the advanced search features and configure custom dashboards in Odoo. By the end of the chapter, you will be able to create and save custom searches to reuse later and add search results to dashboards.

Chapter 11, *Building a Website with Odoo*, is dedicated to exploring Odoo's powerful new website-building platform. At the beginning of the chapter, we look at what a Content Management System (CMS) is and some of the other popular website-building platforms. We follow along with Odoo's website-building tutorial and then look at the features that can be used to promote your website right from within Odoo.

Chapter 12, *Implementing E-Commerce with Odoo*, builds on the previous chapter by adding a fully functioning online shopping cart to the website. We look at how to publish products to the website and the various options for changing their appearance. Midway through the chapter, we cover product variants that add additional flexibility to how you manage your products within Odoo. Finally, we conclude by examining how to set up a payment processor to take payments online through PayPal.

Chapter 13, *Customizing Odoo for Your Business*, explains how to enter developer mode to make a variety of custom changes to Odoo. We walk through the steps to add fields to the sales order form, and then include the fields in tree views for sorting and reporting. From here, we get into advanced configuration topics to better customize Odoo for your specific business requirements.

Chapter 14, *Modifying Documents and Reports*, goes over the basic reporting options that allow you to change the footer and access the QWeb source for your internal and external reports. Next, we see how to modify existing reports in Odoo.

Chapter 15, *Discovering Custom Odoo Modules*, introduces the process of developing custom solutions in Odoo. We build on what we have learned about customizing Odoo and create a module that will persist in our custom field and views within our module.

Chapter 16, *Comparative Analysis of Community versus Enterprise Editions*, provides information that will help you to decide which version of Odoo, Community Edition or Enterprise Edition, is right for you.

Appendix A, *Locating Additional Odoo Resources*, includes a list of resources that can help in extending your knowledge when it comes to supporting an Odoo installation.

To get the most out of this book

This book is based on the most stable version of Odoo, that is, Odoo 12. You should have Odoo version 12 installed on your system. It can be downloaded from <https://github.com/odoo/odoo>. Most often, it is installed in VMware or in the cloud, such as with AWS:

- Chapter 1, *Setting Up Odoo 12*, provides the basic Odoo installation steps for both Windows and Ubuntu.
- Odoo now provides quality documentation on how to install Odoo on a variety of platforms, <https://www.odoo.com/documentation/12.0/setup/install.html>.

Once Odoo is installed, no other software installation is required for the book.

To get the most out of this book, you should understand basic business operations. For example, you should know the purpose of a sales order and a purchase order. You should also have basic computing skills and be able to understand filesystems and install the software. For the more advanced customization topics in the book, you should have basic knowledge of databases and programming concepts.

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: https://static.packt-cdn.com/downloads/9781789536898_ColorImages.pdf.

Conventions used

There are a number of text conventions used throughout this book.

CodeInText: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "First, we define `Orders` to give us access to the `sale.order` model."

A block of code is set as follows:

```
daterequired = fields.Date('Date Required', required=True)
rush = fields.Boolean('Rush Order')
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
from odoo import http

class Webrushorders(http.Controller):

    @http.route('/orders/rush/', auth='public')
    def index(self, **kw):
```

Any command-line input or output is written as follows:

```
cd /usr/lib/python3/dist-packages/odoo/addons
```

Bold: Indicates a new term, an important word, or words that you see onscreen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Using the quick **Create and Edit...** option, you can add your finished products directly when creating a manufacturing order."



Warnings or important notes appear like this.



Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at customercare@packtpub.com.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit <https://www.packtpub.com/support/errata>, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit packt.com.

1 Setting Up Odoo 12

Odoo is a powerful business application framework with a complete suite of enterprise business applications. When you first install Odoo, the only functionality you will have is a limited messaging options between users. From there, Odoo allows you to install the applications you need, as you need them. This flexibility makes Odoo much more accessible than many business software solutions.

The release of Odoo 12 has introduced quite a number of impressive new features and enhancements. Among them are improved layout for mobile devices, a Documents app (a full-featured document management system), keyboard navigation and shortcuts, mass actions, **Optical Character Recognition (OCR)** to import bills emailed from vendors, a new report designer (Odoo Studio), data enrichment that auto-completes company information, **Customer Relationship Management (CRM)** email templates, automated lead generation with user-defined rules, enhanced bank synchronization, additional payment acquirers, a pop-up chatbox, dashboards that more closely resemble the familiar Google Analytics, and vastly improved overall system speed.

In this chapter, we will get started on working with Odoo by covering the installation and the basics of setting up an Odoo database.

The topics we will cover include the following:

- A comparison of Community and Enterprise Editions
- Setting up a trial company on Odoo Online
- Installing Odoo on Windows and Ubuntu
- Troubleshooting and configuring your installation

Community and Enterprise Editions of Odoo

Beginning with Odoo 9, Odoo SA started releasing two versions of Odoo—a **Community Edition** and an **Enterprise Edition**. The Community Edition is free, open source, and primarily supported by the Odoo community. Odoo Enterprise, while also open source, requires a license based on the number of users. More recently, Odoo pricing for the Enterprise version has been varied depending on region.

Odoo Enterprise offers an alternative (and arguably better) user interface. Additionally, there is better functionality in some applications, as well as support by SA and, perhaps most critical of all, migration to the new releases of Odoo. More information on the differences between Odoo Community and Odoo Enterprise Editions is available in Chapter 16, *Comparative Analysis of Community versus Enterprise Editions*.

This book primarily targets the Community Edition of Odoo.

Getting started with Odoo Online

Not long ago, nearly all companies kept their primary information systems in-house. This approach not only requires a lot of capital expense in purchasing servers and software licenses, but also creates a lot of responsibilities and risks in backing up data and ensuring business continuity. Today, more and more companies are choosing to host their business applications in online networks commonly called the cloud. Odoo allows you the flexibility of both options—either hosting on your own hardware, or utilizing Odoo's online software services.

Odoo also offers developers access to Odoo.sh, a cloud-based, GitHub-enabled staging server for beta-testing customized apps and deploying them to a managed production server. This service is available to those who have a license for the Odoo Enterprise Edition and would like to have it hosted like Odoo Online but require customized features. For further information about this Odoo.sh, visit <https://www.odoo.sh/features>.

Taking advantage of Odoo Online instant access

The best thing about accessing Odoo Online is that you can jump in and start using the software right away. You don't have to decide what **operating system (OS)** to use, and you don't have to install any software at all. Just enter the URL into your web browser and you are ready to get started.

Another added benefit of taking this approach is that you will verify that your web browser is up to date and compatible with the latest version of Odoo. So, even if you intend to install Odoo on your own hardware, it is still worth taking a minute to test out the Online Trial version of Odoo. You can expect to put a great deal of time into determining which Odoo applications are right for your company.

Taking a few hours to use the Odoo Online version is time well spent, and you can put off installing Odoo until you are more certain it is the right software for your business.



Use the Odoo Trial Edition to verify browser compatibility with any older machines.

Odoo browser requirements

Odoo is designed to run on a variety of modern web browsers. Supported browsers include the following:

- Google Chrome (recommended)
- Firefox
- Internet Explorer
- Safari



Macintosh users will need to make sure they are running **macOS X** or above. Users running older Macintosh systems are currently having difficulties running Odoo version 7. Also, in my experience, Google Chrome tends to offer the best experience in working with Odoo. Firefox is also often recommended by others in the Odoo community.

Odoo mobile phone and tablet support

Beginning with Odoo 8, Odoo has included native support for mobile phones and tablets. As with previous Odoo upgrades since, mobile support has been further improved in Odoo 12. Menus are designed to flow and format properly. The new website application even includes a preview within the portal administration to emulate how the site would appear on a mobile phone. While you still suffer many of the limitations that come with a small screen size, the applications are functional and this makes it even easier for developers to create mobile Odoo applications.

Odoo's mobile-application support covers both the Android and Apple iOS platforms. Note, however, that these mobile apps will only be able to connect to Odoo Enterprise; Odoo Community installations are not supported. Make sure you thoroughly test all processes that you intend to implement for your business, and test them on both desktop and mobile platforms. The smaller screen sizes might make some data unreadable or very awkward to work with.

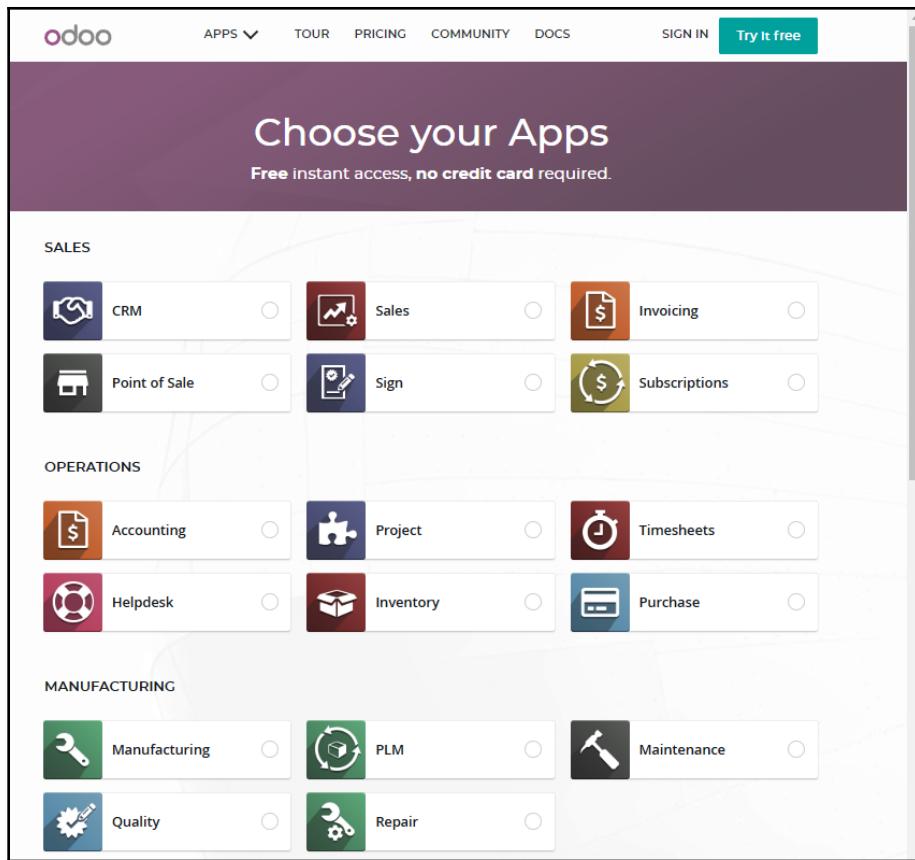
Accessing the Odoo free online trial

Accessing the online trial version of Odoo Online could not be simpler. Just open up your browser and navigate to <https://www.odoo.com/trial>.



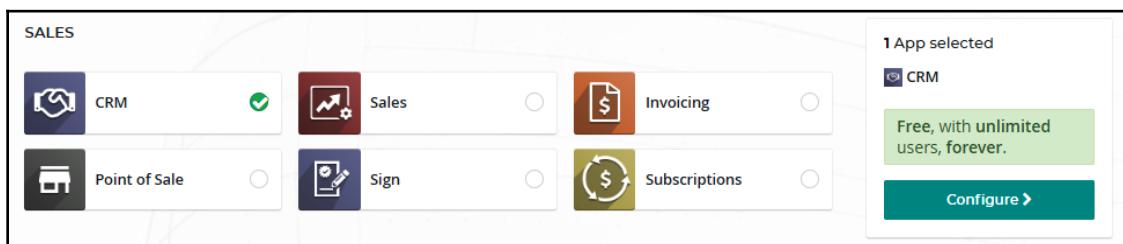
Please be aware that Odoo Online's trial is the Enterprise Edition of Odoo. While very similar to the Community Edition, the interface will be slightly different.

You will then be prompted to choose one of Odoo's primary business applications, as shown in the following screenshot:



Don't worry too much, because you can add more applications later.

Clicking on the appropriate button for the application immediately begins installing your own unique Odoo instance. For our example, let's go ahead and install the CRM application:



For installing the CRM application, you will need to perform the following steps:

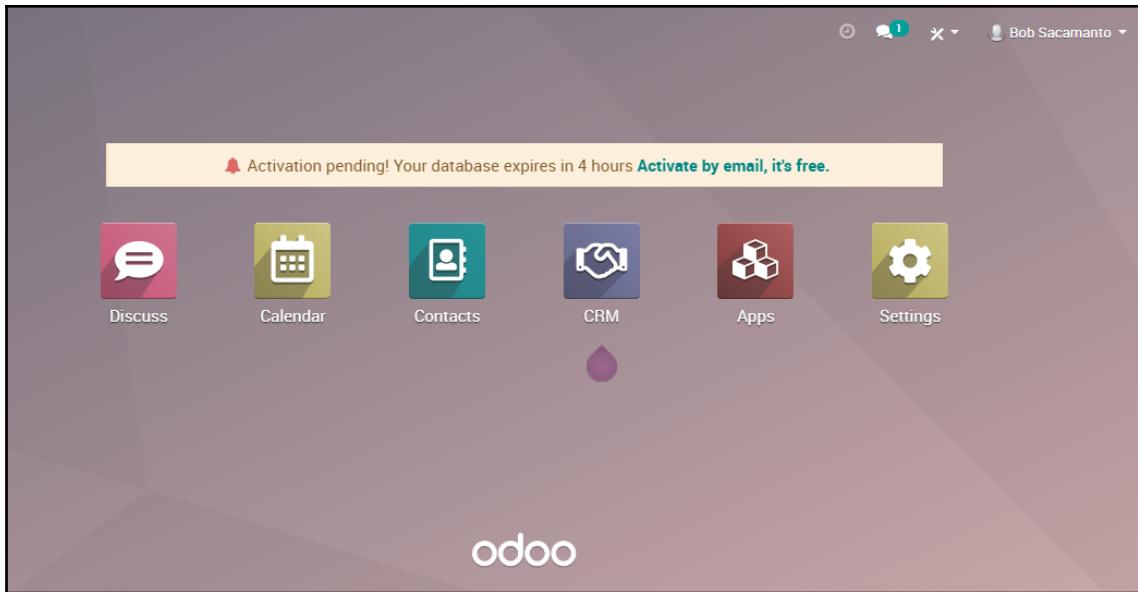
1. Click the radio button for CRM.
2. Now, click **Configure**, and you will be taken to a simple sign-up page.
3. Fill out the **Get Started** form and click **Start Now**, as shown in the following screenshot:

The screenshot shows the Odoo Get Started page. At the top, there is a navigation bar with links for APPS, TOUR, PRICING, LEARN, COMMUNITY, SIGN IN, and a prominent green 'Try it free' button. The main heading is 'Odoo | Get Started' with the subtext 'Instant access.' Below this, there are several input fields:

- 'First and Last Name': Bob Sacamanto
- 'Email': bobsacamanto@example.com
- 'Company Name': OdooClass
- 'Phone Number': +1312323333
- 'odooclass.odoo.com' (with a link icon)
- 'Country': United States
- 'Language': English
- 'Company size': 5 - 20 employees
- 'Primary Interest': Offer Odoo services to other compa (with a dropdown arrow)

A note at the bottom states: 'By clicking on Start Now, you accept our Subscription Agreement and Privacy Policy'. At the very bottom, there are two buttons: 'Change apps selection' and a large green 'Start Now >' button.

Be patient, as it can take thirty seconds or longer for the servers to build the database and bring up the starting page. When the installation is complete, Odoo automatically signs you in so you can begin trying out the software. The goal of this approach is to get users to start using the software right away, and avoid having to fill out lengthy forms or create logins and passwords to begin using the software. It really is just one click, and you have your own version of Odoo to evaluate:



The screen now contains the Odoo dashboard for the Enterprise Edition. Please be aware that this will look a bit different than the Community Edition of Odoo, even though they are very similar in terms of functionality.

We will discuss the CRM application in detail in [Chapter 3, Exploring Customer Relationship Management in Odoo 12](#).

Take a few minutes to look around in Odoo to get familiar with the interface. You don't have to worry about breaking anything or doing anything wrong. If you run into problems or get confused, just close your web browser and try again.



This is a demonstration and will only last for a four-hour session. If you close your browser, you will lose your setup and have to start over again.

Continuing to use the trial version of Odoo

At the very top of the Odoo application, just under the address bar in the browser, you will see a message that informs you of how much longer your trial version of Odoo will run before you need to register. Also, remember that it is possible to lose this instance of Odoo before the time runs out, as seen here:

 Activation pending! Your database expires in 4 hours [Activate by email, it's free.](#)

Clicking on the **Activate by email** link will send a message to the email address that you provided, containing a link to activate your Odoo database. Currently, you may use Odoo for free as long as you only install one application.



Odoo SA has experimented quite a bit with different trial terms, so do not be surprised if the trial options are slightly different by the time you are reading this.

Subscribing to Odoo

You can use one Odoo application for free without subscribing. If you wish to install more applications (as you likely will), you must subscribe to Odoo in order to keep using their Enterprise cloud-hosted version of the software.

For each user, the current pricing is \$24 per month. At the time of this writing, there is a \$4-per-month user discount, thus reducing the total per-month cost to \$20. Each application you use will also incur a monthly cost that varies for each respective application.

The following is the **Odoo Online Pricing** calculator for October 2018 in the United States with one user and three applications:

Odoo Pricing

Choose the number of Users

1	Users x \$25.00 USD/Month	\$30.00 USD/Month
---	---------------------------	-------------------

Choose your Apps

CRM	Invoicing	Sales %	Annually	Monthly
\$30.00 USD / month	\$15.00 USD / month	\$15.00 USD / month	1 Users	\$30.00 USD
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Website	eCommerce %	Point of Sale %	User Discount (*)	-\$5.00 USD
\$30.00 USD / month	\$15.00 USD / month	\$30.00 USD / month		
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Accounting %	Project	Inventory	3 Apps	\$60.00 USD
\$30.00 USD / month	\$30.00 USD / month	\$45.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Total / month	\$85.00 USD
Manufacturing %	Purchase %	Timesheet %		
\$60.00 USD / month	\$15.00 USD / month	\$15.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Email Marketing	Expenses %	Events %		
\$15.00 USD / month	\$15.00 USD / month	\$15.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Leaves	Recruitment	Appraisal %		
\$15.00 USD / month	\$15.00 USD / month	\$15.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Subscription %	Sign	Equipment		
\$30.00 USD / month	\$30.00 USD / month	\$15.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Quality Control %	Studio	Helpdesk		
\$30.00 USD / month	\$90.00 USD / month	\$30.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Maintenance (Manufacturing) %	PLM (Manufacturing) %	Appointment Scheduling %		
\$30.00 USD / month	\$30.00 USD / month	\$15.00 USD / month		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

(*) New customers get a discount on the initial number of users purchased. (\$25.00 USD instead of \$30.00 USD).

Try now - 15 days Free trial

Buy now

You can locate the **Odoo Online Pricing** page at <https://www.odoo.com/pricing-online>.



Odoo Online is priced for employees that use the applications. You are not charged for customers or suppliers that access Odoo through the web portal.

Depending on your requirements, an Odoo subscription might be a good decision. Installing and maintaining an Odoo installation takes a degree of expertise and has risks for production systems. You must maintain adequate disaster recovery procedures in case of server crashes or hard drive failures. There are also complexities in applying bug fixes and migrating to newer versions of Odoo. This book will help you with many of these tasks. It can be quite convenient to have an Odoo subscription so that you can focus on the functional, rather than the technical, aspects of working with Odoo.

Using Odoo without subscription fees

One of the greatest advantages of Odoo is that Odoo Community is both open source and can be used without paying any license fees. If you choose not to pay the subscription fee, do not fear! The remainder of this chapter will assist you with installing Odoo on your own hardware.

Getting to know the Odoo architecture

Setting up and managing an Odoo installation will require a basic understanding of the components that make up Odoo. Every business system has a set of technologies and underlying software platforms that are required for the system to function. Fortunately, unless you plan to customize Odoo, you only need to understand the very basics of the Odoo architecture to complete a successful installation.



In this book, we provide a basic overview of the Odoo architecture. If you wish to get more detailed documentation on the Odoo architecture and technical documentation, visit <https://www.odoo.com/documentation/12.0/>.

Introducing the PostgreSQL database

Like most ERP systems, Odoo has specific database requirements: in this case, PostgreSQL. PostgreSQL is an open source, cross-platform **Object Relational Database Management System (ORDMS)**. While not popular on the scale of Microsoft SQL Server or MySQL, PostgreSQL is an enterprise-class database server with many advanced features. In fact, PostgreSQL stacks up very well against far more expensive databases such as Microsoft SQL Server and Oracle Database.

PostgreSQL runs on every major OS. For most Odoo installations, Ubuntu is the OS of choice. However, PostgreSQL will also run quite well under other versions of Linux, Microsoft Windows, and even macOS X.

You can learn more about PostgreSQL at <http://www.postgresql.org/>.

Writing code with Python

The primary programming language of Odoo is Python. Like the other technologies underlying Odoo, the Python language is open source and runs on all of the major contemporary OS. It is an extremely popular programming language, which makes it very easy to find resources to help you to get started.

Starting in Odoo 11, Odoo began using Python 3.5 instead of Python 2.7.

You can learn more about the Python programming language at <http://python.org/>.

Following the Model-View-Controller design

Odoo is built upon a **Model-View-Controller (MVC)** architecture. One of the primary goals of this architecture is to separate the visual display of the information from the business rules and management of the underlying data. For example, if you need to change the way data is organized in the model, it is desirable not to have to make dramatic changes to how you view the data. This is true for maintaining flexibility in viewing data. Today, it is common to have many different client applications sharing the same underlying data.

Designing models

The model is essentially the data that makes up your Odoo installation, which is stored in the PostgreSQL database. Odoo is unique in that database structures are typically defined by the Odoo modules at the time they are installed. The Odoo framework takes the model definitions and automatically creates the necessary table structures inside of the PostgreSQL database. Furthermore, a web interface in Odoo allows administrators to easily extend the Odoo data model in a variety of ways without having to modify the Odoo source code.

Rendering views

Each view in Odoo is defined in XML documents. The Odoo framework is responsible for rendering these view files in a web browser. Alternative views can be built to render Odoo functionality upon other platforms such as mobile devices.

Authoring controllers

The controller component of the architecture is where the business logic and workflow rules of the Odoo application are applied. The controller components in Odoo are written in Python code and stored as objects in Odoo modules.

Choosing your installation OS

In this section, we will discuss some of the advantages and disadvantages of choosing Ubuntu or Windows for your first Odoo installation.

Choosing a Microsoft Windows Odoo installation

For the most part, Ubuntu has been the platform of choice for most Odoo installations. However, there are some reasons why you might choose to run Odoo under a Windows installation.

Some of you, after buying this book, might have already jumped ahead and installed Odoo on your Microsoft Windows computer. So, for you go-getters, that working installation of Odoo might function just fine for researching and testing its features. Often, the Windows all-in-one installer provides a simple method to get Odoo up and running in a snap on your hardware. Basically, you do not have to install a new OS.

Learning Ubuntu is not required

If you are familiar with Windows and have no Ubuntu experience, you might get going a little faster by sticking with a Windows installation for your first setup. Downloading and installing modules and making changes to configuration files will be much easier if you are familiar with the OS.

Introducing Ubuntu

While Microsoft Windows does not really need an introduction, it is probably worth giving a brief introduction to Ubuntu. In short, Ubuntu (pronounced *oo-BOON-too*) is a very popular open source OS based on the Linux kernel. It has enjoyed increasing popularity because it is easy to install and very stable. Ubuntu can be installed either as a server OS without a graphical interface, or as a desktop OS with a graphical interface, that closely resembles Windows.

You can learn more about the Ubuntu OS, and why it is so popular, at <http://www.ubuntu.com/>.

Choosing an Ubuntu Odoo installation

It is generally accepted that Ubuntu is the recommended OS for running a production installation of Odoo. There are several reasons why this is true:

- **Ubuntu is the primary target platform:** While Odoo is released for Windows and is still well supported, the Ubuntu installation continues to be favored. The development team for Odoo works primarily with Ubuntu for bug fixes and platform releases. It can be expected that, for the most part, Odoo development will be optimized around Ubuntu, not Windows or macOS.
- **Ubuntu is open source:** Installing Odoo on any Windows OS is going to require a license from Microsoft. While using Odoo on your Windows PC or macOS is a viable and perhaps desirable solution for testing and development, it is unlikely you will want to run Odoo on a Windows desktop system for any production environment. Why? Well, this requires Windows Server, which has much higher license costs than desktop editions. With an Ubuntu installation, you get an entirely open source and virtually cost-free solution.
- **Ubuntu has additional scalability options:** It is possible to configure a more scalable solution under Ubuntu than what you can currently configure under Microsoft Windows Server. Specifically, multiprocessing mode is not available, at least with the default Odoo release, under the Windows OS.
- **Ubuntu has strong community support for Odoo:** The fact is that a vast majority of the production installations of Odoo are running under Ubuntu. When you run into trouble or management issues with your Odoo installation, you may find it easier to get assistance if you are running an Ubuntu installation.

Choosing another OS option for Odoo

Although this book will focus on Window and Ubuntu installations, you do have several other options. In the past, Odoo has been deployed under a variety of Linux distributions and even macOS. There are also many community members actively developing client frontends for mobile platforms such as Google's Android OS.

Understanding Odoo releases

When deploying an Odoo system, it is important to understand the various Odoo versions, as well as the release and upgrade policies. There is currently one major release each for versions 10.0, 11.0, and 12.0, as well as a master branch that is the latest development version, which will become Odoo version 12.0. The stable versions are the standard support versions of Odoo, and typically the ones you should choose to install for most situations. The master version is the development version and will often contain bugs and unfinished features. This is primarily downloaded by developers and those who wish to get a look at the latest features.

Upgrading Odoo

The goal of the Odoo development team is to release two stable version upgrades each year. Odoo further labels some stable versions as **Long-Term Support (LTS)** versions. These releases are supported by Odoo for those that have an Odoo Enterprise support contract. For any production environment, it is smart to choose an LTS version. Most importantly, installing an LTS release of Odoo will make bug fixes and patches much easier to implement.



At the time of this writing, the most recent stable LTS version is version 12.0.

Installing Odoo on Windows OS

We begin our installation by locating the packages that are currently available to install. You can find the current list at <http://nightly.odoo.com/>. The following screenshot is the **Odoo Nightly builds** page, which is the jumping-off point for downloading the source files for installation:

The screenshot shows a web browser window with the URL nightly.odoo.com in the address bar. The page title is "Odoo Nightly builds". Below the title, there is a section titled "Builds" which contains a paragraph about nightly builds and their support timeline. A list of package versions is displayed in a sidebar:

- 12 (stable) - Community Edition**: Odoo 12 was released in October 2018 and is supported until Odoo 15 - the recommended version.
- 11 (stable) - Community Edition**: Odoo 11 was released in October 2017 and is supported until Odoo 14 - the recommended version.
- 10 (stable) - Community Edition**: Odoo 10 was released in October 2016 and is supported until Odoo 13.
- master (dev) - Community Edition**: This branch contains experimental features - for testing only.

At the bottom of the sidebar, there is a link to "Check out our [installation](#) and [deployment](#) guides."

The examples and case studies in this book use Odoo 12.0. This means you should select the 12.0 (stable) version of Odoo to download. You can navigate directly to the Odoo 12.0 downloads at <http://nightly.odoo.com/12.0/nightly/>, and you should see a screen similar to the following screenshot:

The screenshot shows a web browser window with the URL nightly.odoo.com/12.0/nightly/ in the address bar. The page title is "Index of /12.0/nightly/". Below the title, there is a table listing files:

.. /		
deb/	30-Oct-2018 08:05	-
exe/	30-Oct-2018 08:01	-
rpm/	30-Oct-2018 08:08	-
src/	30-Oct-2018 08:10	-



It is entirely possible that Odoo will change the URL as new versions are released. To best follow the examples in this book, download a 12.x installation of Odoo.

Windows installations of Odoo use EXE packages. Click on the `exe/` directory to get the list of downloads that are available.

Naturally, the specific download packages are going to change on a nightly basis.

The latest version of the stable LTS release will contain the most current Odoo build, with bug fixes included, and will appear at the bottom of the list. By the way, the upload dates you'll see are in **Coordinated Universal Time (UTC)**, and therefore might be many hours ahead of your time zone, especially if you live in the Western Hemisphere.

Click the `LATEST.EXE` file to download the most current build to your computer.

Performing an all-in-one Odoo installation on Windows

Installing Odoo using the all-in-one package is very simple. After the package has finished downloading, double-click on the `.exe` file to begin the installation wizard.

The first screen will prompt you to select the language for your install.

After you have selected the language and clicked on **OK**, the wizard will continue with the installation. From here, everything will continue like a normal installation on Windows.



I highly recommend that you choose the **Custom Install**, so you can select the directory for installation. The default directory name contains the lengthy build number, making it rather difficult to work with in Command Prompt.

Configuring Postgres on Windows

During the installation, you will be asked to provide information for the PostgreSQL connection, as in the following screenshot:

The image shows the Odoo setup wizard interface. At the top is the Odoo logo. Below it, a message says "Odoo is up and running! Fill out this form to create a new database. You will install your first app afterwards." The form includes fields for "Database Name", "Email", and "Password". There is also a dropdown for "Language" set to "English" and another for "Country". A checkbox labeled "Load demonstration data (Check this box to evaluate Odoo)" is present. At the bottom are two buttons: a blue "Create database" button and a link "or restore a database".

It is recommended that you change the default username and password for security purposes. These values will be written into the Odoo configuration file. The username and password provided will be the administration credentials for the PostgreSQL database, so be sure to remember them.

After the wizard is complete, if you leave **Start Odoo** checked and then click on **Finish**, Odoo should open up in a new tab of your default browser.

If Odoo fails to launch, you can look at the *Troubleshooting and Odoo Management Tips* section later in this chapter for solutions to some of the problems commonly encountered during installation.

Installing Odoo on Ubuntu

This book will walk you through the installation procedure for Odoo on Ubuntu using the latest all-in-one nightly package. Depending on your Ubuntu installation and how you want to work with Odoo, there are alternative installation methods.

At the time of writing, Odoo is most commonly installed on Ubuntu version 18.04.

Modifying the sources.list file

Installing Odoo in Ubuntu is easy when you use the Debian repository. In order to perform these operations, you may have to be the root user. If you have an account that has the permissions to do so, you can temporarily switch to the root user by opening a Terminal window and entering the following:

```
sudo -s
```

For better security, and to guarantee you are installing the correct package, Odoo now signs their distributions. The following command adds the correct key to your Ubuntu installation so that it will recognize the Odoo package:

```
wget -O - https://nightly.odoo.com/odoo.key | apt-key add -
```

Next, we want to add the distribution to the `/etc/apt/sources.list` file with the following command line:

```
echo "deb http://nightly.odoo.com/12.0/nightly/deb/ ./" >>
/etc/apt/sources.list.d/odoo.list
```

This installs the package.

You can start the installation process by entering these commands into a Terminal window:

```
sudo apt-get update
sudo apt-get install odoo
```

The Odoo packages will first be downloaded and then installed. This is an all-in-one installation and should set up all of the necessary packages, PostgreSQL, and library dependencies required to run Odoo.

By default, the deb installation will place the source in the `/usr/lib/python3/dist-packages/odoo` directory.

Testing your Odoo installation

Point your browser to `http://localhost:8069` and you should see the Odoo database creation page appear.

Troubleshooting and Odoo management tips

As far as ERP installations go, Odoo is typically very easy to install. Unfortunately, it is possible for an installation to fail for a variety of reasons. In this next section, we will discuss some of the most common installation issues and provide some troubleshooting tips for diagnosing problems with an Odoo installation.

Checking your browser destination

If you have followed the default installation, then your Odoo installation should be accessing Odoo at `http://localhost:8069`.

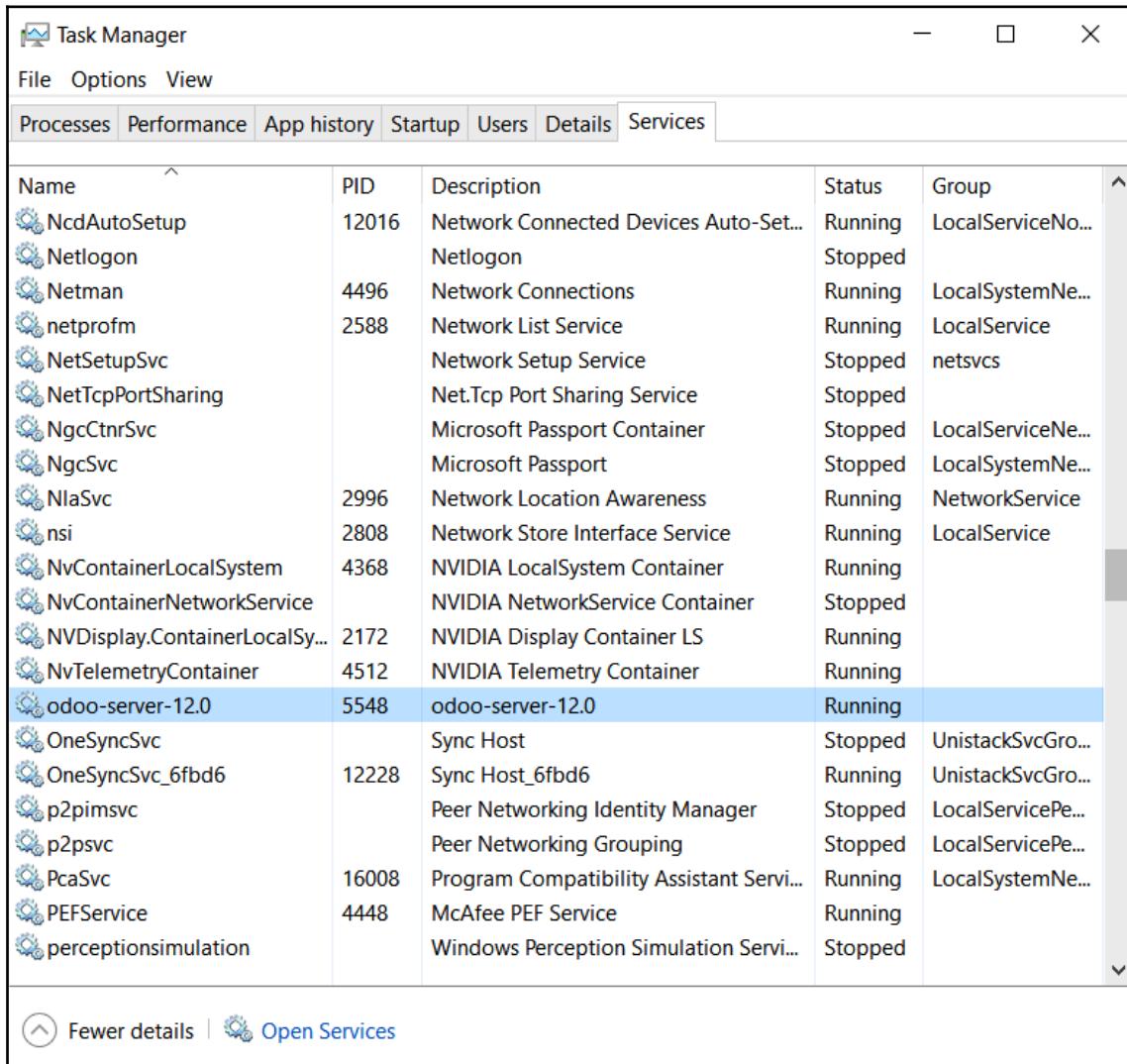
Make sure the URL is exactly as you see it here. If you changed the port number during installation, make sure you change the port in the URL.

Verifying that the Odoo service is running

If you are unable to pull up Odoo in the browser and instead see the error message **ERR_CONNECTION_REFUSED** or **localhost refused to connect**, it might be good to verify that the Odoo service is running.

Checking for the Odoo services running in Windows

Pull up the task manager and go to the **Services** tab, then look for `odoo-server-12.0`. The status should be **Running**, as shown in the following screenshot:



The screenshot shows the Windows Task Manager with the 'Services' tab selected. The table lists various Windows services, including the 'odoo-server-12.0' service which is highlighted in blue and has a status of 'Running'. Other services listed include NcdAutoSetup, Netlogon, Netman, netprofm, NetSetupSvc, NetTcpPortSharing, NgcCtnrSvc, NgcSvc, NlaSvc, nsi, NvContainerLocalSystem, NvContainerNetworkService, NVDisplay.ContainerLocalSy..., NvTelemetryContainer, OneSyncSvc, OneSyncSvc_6fb6, p2pimsvc, p2psvc, PcaSvc, PEFSvc, and perceptionsimulation.

Name	PID	Description	Status	Group
NcdAutoSetup	12016	Network Connected Devices Auto-Set...	Running	LocalServiceNo...
Netlogon		Netlogon	Stopped	
Netman	4496	Network Connections	Running	LocalSystemNe...
netprofm	2588	Network List Service	Running	LocalService
NetSetupSvc		Network Setup Service	Stopped	netsvcs
NetTcpPortSharing		Net.Tcp Port Sharing Service	Stopped	
NgcCtnrSvc		Microsoft Passport Container	Stopped	LocalServiceNe...
NgcSvc		Microsoft Passport	Stopped	LocalSystemNe...
NlaSvc	2996	Network Location Awareness	Running	NetworkService
nsi	2808	Network Store Interface Service	Running	LocalService
NvContainerLocalSystem	4368	NVIDIA LocalSystem Container	Running	
NvContainerNetworkService		NVIDIA NetworkService Container	Stopped	
NVDisplay.ContainerLocalSy...	2172	NVIDIA Display Container LS	Running	
NvTelemetryContainer	4512	NVIDIA Telemetry Container	Running	
odoo-server-12.0	5548	odoo-server-12.0	Running	
OneSyncSvc		Sync Host	Stopped	UnistackSvcGro...
OneSyncSvc_6fb6	12228	Sync Host_6fb6	Running	UnistackSvcGro...
p2pimsvc		Peer Networking Identity Manager	Stopped	LocalServicePe...
p2psvc		Peer Networking Grouping	Stopped	LocalServicePe...
PcaSvc	16008	Program Compatibility Assistant Servi...	Running	LocalSystemNe...
PEFSvc	4448	McAfee PEF Service	Running	
perceptionsimulation		Windows Perception Simulation Servi...	Stopped	

[Fewer details](#) | [Open Services](#)

This screenshot shows an example of the `odoo-server-12.0` service successfully running on Windows.

Additional Odoo troubleshooting steps for Windows can be found at https://doc.Odoo.com/install/windows/server/complementary_install_information/.

Checking for Odoo services running in Ubuntu

In Ubuntu, you can locate the Odoo service by running the following command in a Terminal window:

```
ps aux | grep Odoo
```

You will then see the Odoo service listed if it is running:

```
root@ubuntu:~# ps aux | grep odoo
odoo    14341  1.0 11.1 925300 111756 ?      Sl   11:40   0:02 /usr/bin/python /usr/bin/odoo --config /etc/odoo/odoo.conf --logfile /var/log/odoo/odoo-server.log
postgres 14489  0.0  1.1 308992 11016 ?      Ss   11:42   0:00 postgres: odoo postgres [local] idle
root     14539  0.0  0.1 21292  1008 pts/17  S+   11:45   0:00 grep --color=auto odoo
root@ubuntu:~#
```

Now, let's move on to the next section.

Starting and stopping Odoo services in Ubuntu

When managing an Odoo server, one of the most common tasks you will find yourself performing is starting and stopping the Odoo service. Odoo allows you to start and stop the service with a command switch.

To start the service, use the following command:

```
sudo /etc/init.d/odoo start
```

To stop the service, use the following command:

```
sudo /etc/init.d/odoo stop
```

Finding the primary Odoo log file

Odoo writes many messages, warnings, and error messages to a log. Often, when troubleshooting problems, this log file is valuable in determining what action you should take. In a default installation, the log file is located at `/var/log/odoo/odoo-server.log`.

The log is especially valuable for locating problems you may have when installing new modules.

Modifying the Odoo configuration file

The Odoo framework allows you to specify a configuration file for your installation. By default, this file is located at `/etc/odoo/odoo.conf`.

Using this file, you can change many of the attributes of Odoo.

Changing port numbers

By default, Odoo runs on port 8069. For many installations, the default port will work fine. There are situations, however, when it can be useful to change this default port. One common scenario would be the need to run more than one version of Odoo. Multiple installations cannot run on port 8069, so you will need to modify the port on all installations following the first one. Sometimes, there are security reasons behind changing ports, as many hackers are aware of the default ports people use.

Fortunately, changing the default port number is easy.

Simply specify the following:

```
Port=[port]
```

For example, `Port=8059` will change the default port for the web client to port 8059.

Accessing the database management tools

Odoo offers database management tools that can be accessed easily through your web browser. This makes it easy to create, backup, and even delete databases, all through a web interface. While there are sometimes links available on the login page that will take you to these tools, it is possible that, when installing some applications, such as the website builder, you will not find a link easily.

To access the database management tools, use the following path:

```
[ServerAddress] : [port] /web/database/manager
```

Changing the admin password

As mentioned earlier, by default, Odoo sets the password for its operations to `admin`. To secure your server, it is necessary to change this password in your configuration file:

```
Admin_password=[your password]
```

Also, be careful not to start up your Odoo server from the command line without specifying an alternative password or the path to the configuration file. If you do, you leave the instance open with the default password.

Finding additional resources on installing Odoo

Installing and configuring Odoo can quickly become a very complex task that is outside the scope of this book. In the [Appendix A, Locating Additional Odoo Resources](#), of this book, you will find links to additional resources that can assist you with installing Odoo.

Summary

In this chapter, we saw how easy it was to get started using Odoo Online. We discussed how to set up a trial company, and the basics for creating a database and installing your first module. If you choose not to use the online services, you likely found the topics on installing Odoo on Windows or Ubuntu helpful. Finally, we discussed various methods of troubleshooting and configuring Odoo.

In the next chapter, we will begin to jump into our first real business applications in Odoo. You will get introduced to our real-world case study and set up the basic configuration for the company. We will walk you through setting up your first product and, finally, creating and printing your first sales order.

2

Installing Your First Application

We have learned about the various applications that Odoo has to offer and how you can install Odoo on your own system. Before the release of Odoo 8, most users were focused on ERP and finance-related applications. Now, Odoo 12 has added several important applications that allow companies to use Odoo in much greater scope than ever before. For example, the website builder can be installed to quickly launch a simple website for your business—a task that typically would need to be accomplished with a content management system such as WordPress.

Despite all of the increasing options available in Odoo, the overall process is the same. We begin by looking at the general business requirements and decide on the first set of applications that we wish to implement. After understanding our basic objectives, we will create an Odoo database and configure the required company information.

Next, we begin exploring the Odoo interface for creating and viewing information. We will see just how easy Odoo is to use by completing an entire sales order workflow. We will finish up this chapter by reviewing some of the more advanced sales order configuration options.

Topics that we will cover in this chapter include the following:

- Defining requirements and reasons to take a modular approach
- Adding a password-protected database to our installation
- Installing and configuring the Sales Management module
- Using interface features to view, edit, and find information
- Adding our first product to sell
- Writing a sales order
- Creating invoices

Gathering requirements

Setting up an optimal Odoo system is no easy task. Many companies get into trouble believing that they can just install the software and throw in some data. Inevitably, the scope of the project grows and what was supposed to be a simple system ends up being a confusing mess. Fortunately, Odoo's modular design will allow you to take a systematic approach to implementing Odoo for your business.

Implementing Odoo using a modular approach

The bare bones installation of Odoo simply provides you with a limited messaging system. To manage your Odoo implementation, you must begin with planning the modules with which you will work first. Odoo allows you to install just what you need now and then install additional Odoo modules as you better define your requirements. It can be valuable to take this approach when you are considering how you will implement Odoo for your own business.



Don't try and install all of the modules and get everything running all at once. Instead, break down the implementation into smaller phases.

Introducing Silkworm – our real-world case study

To best understand how to work with Odoo, we will build our exercises around a real-world case study. Silkworm is a midsized screen printer that manufactures and sells t-shirts as well as a variety of printing projects. Using Odoo's modular design, we will begin by implementing the **Sales Order** module to set up the selling of basic products. In this specific case, we will be selling t-shirts. As we proceed through this book, we will continue to expand the system by installing additional modules.



When implementing Odoo for your organization, you will also want to create a basic requirements document. This information is important for the configuration of the company settings in Odoo and should be considered an essential piece of documentation when implementing an ERP system.

Creating a new database in Odoo

If you have installed Odoo on your own server, now you will need to create a database. The necessary tables and fields will be added to the database you specify as you install additional applications to Odoo.



Odoo Online: If you are using Odoo Online, you will not have access to create a new database and instead will use Odoo's one-click application installer to manage your Odoo installation.

After you have installed a fresh copy of Odoo, you will be prompted automatically to create a new Odoo database:

Create Database	
Database Name	SILKWORM-DEV
Email	admin
Password	**** <input type="button" value="eye"/>
Phone number	123-123-1234
Language	English
Country	United States
Demo data	<input type="checkbox"/>
To enhance your experience, some data may be sent to Odoo online services. See our Privacy Policy .	
<input type="button" value="Continue"/>	

In the preceding screenshot, you can see the Odoo form to create a new database.

Odoo provides basic instructions for creating your database. Let's quickly review the fields and how they are used.

Selecting a database name

When selecting a database name, choose a name that describes the system and that will make clear the purpose of the database. There are a few rules for creating an Odoo database:

- Your database name cannot contain spaces and must start with a number or letter.
- Also, you will need to avoid commas, periods, and quotes.
- Underscores and hyphens are allowed if they are not the first character in the name.

It can also be a good idea to specify in the name whether the database is for development, testing, or production purposes.

For the purposes of our real-world case study, we will use the following database name:

SILKWORM-DEV

We have chosen the `-DEV` suffix as we will consider this a development database that will not be used for production or even for testing.



Take the time to consider what you will name your databases. It can be useful to have standard prefixes or suffixes depending on the purpose of your database. For example, you may use `-PROD` for your production database or `-TEST` for the database that you are using for testing.

Choosing an email address and password

Unlike previous versions of Odoo in which the database was created with an administrator account named `admin`, Odoo 12 uses the email address you provide as your administrator account. This is also known as the **superuser account**.

Please note that, while Odoo asks for you to use an email address, it does not enforce this. You can still use `admin` or an alternative username instead of an email address. In fact, I typically find it easier and more consistent to use `admin` for the main admin user than using a unique email address. Naturally, for a secure production installation, it would be advisable to use strong logins and passwords.

The password you choose during the creation of the database will be the password for this admin account.



Choose any password you wish and click on **Continue** to create the SILKWORM-DEV database.

Specifying a phone number

Now, in Odoo 12, you are prompted to provide a phone number during the creation of the database. You can fill this in if you choose.

Specifying our default language

Odoo offers a variety of language translation features with support for more than 20 languages. All of the examples in this book will use the **English (US)** language option. Be aware that, depending on the language you select in Odoo, you may need to have that language also installed in your base operating system.

Specifying the country

In addition to the language, Odoo asks that you specify the country when you create a new database. This is used to determine the default chart of accounts that will be used for the Odoo installation.

Loading demonstration data

Notice the box reading **Demo Data**. If you mark this checkbox when you create a database, Odoo will preload your tables with a host of sample data for each module that is installed. This may include fake customers, suppliers, sales orders, invoices, inbox messages, stock moves, and products. The purpose of the demonstration data is to allow you to run modules through their paces without having to key in a ton of test data.

For the purposes of our real-world case study in this book, do not load the demonstration data.

Managing databases in Odoo

The database management interface allows you to perform basic database management tasks such as backing up or restoring a database. Often with Odoo, it is possible to manage your databases without ever having to go directly into the Postgres database server. It is also possible to set up multiple databases under the same installation of Odoo. For instance, in the future, you may want to install another database that loads demonstration data and may be used to install modules simply for testing purposes.



If you have trouble getting to the interface to manage databases, you can access the database management interface directly by going to the /web/database/manager path.

Installing the Sales application

After clicking on **Create Database** (depending on the speed of your system), it can take a little time before you are shown a page that lists the available applications:

The screenshot shows the Odoo Apps page with a grid of 20 modules. Each module card includes an icon, the module name, a brief description, an 'Install' button, and a 'Learn More' link. The modules listed are: CRM, Website, Project, Timesheets, Inventory, Barcode, Invoicing, Accounting, Manufacturing, MRP II, Product Lifecycle Management (PLM), Quality, Sales, Studio, Point of Sale, Notes, eCommerce, Purchase, Helpdesk, Employees, Attendances, Recruitment, Expenses, and Dashboards.

Module	Description	Action
CRM	Track leads and close opportunities	Install Learn More
Website	Enterprise website builder	Install Learn More
Project	Organize and schedule your projects	Install Learn More
Timesheets	Track time & costs	Upgrade Learn More
Inventory	Manage your stock and logistics	Install Learn More
Barcode	Barcode scanner for warehouses	Upgrade Learn More
Invoicing	Invoices & Payments	Install Learn More
Accounting	Accounting, Taxes, Budgets, Assets...	Upgrade Learn More
Manufacturing	Manufacturing Orders & BOMs	Install Learn More
MRP II	Work Orders, Planning, Routing	Upgrade Learn More
Product Lifecycle Management (PLM)	PLM, ECOs, Versions	Upgrade Learn More
Quality	Quality Alerts, Control Points	Upgrade Learn More
Sales	From quotations to invoices	Install Learn More
Studio	Create and Customize Applications	Upgrade Learn More
Point of Sale	Tablet POS: shops and restaurants	Install Learn More
Notes	Organize your work with memos	Install Learn More
eCommerce	Sell your products online	Install Learn More
Purchase	Purchase orders, tenders and agreements	Install Learn More
Helpdesk	Track support tickets	Upgrade Learn More
Employees	Centralize employee information	Install Learn More
Attendances	Track employee attendance	Install Learn More
Recruitment	Track your recruitment pipeline	Install Learn More
Expenses	Submit, validate and reinvoice employee expenses	Install Learn More
Dashboards	Build your own dashboards	Install Module Info

This screen lets you select from a list of the most common Odoo modules to install.

There is very little you can do with just an Odoo database with no modules installed. Now, we will install the **Sales** application so we can begin setting up our business selling t-shirts:



Click on the **Install** button to install the **Sales** module.

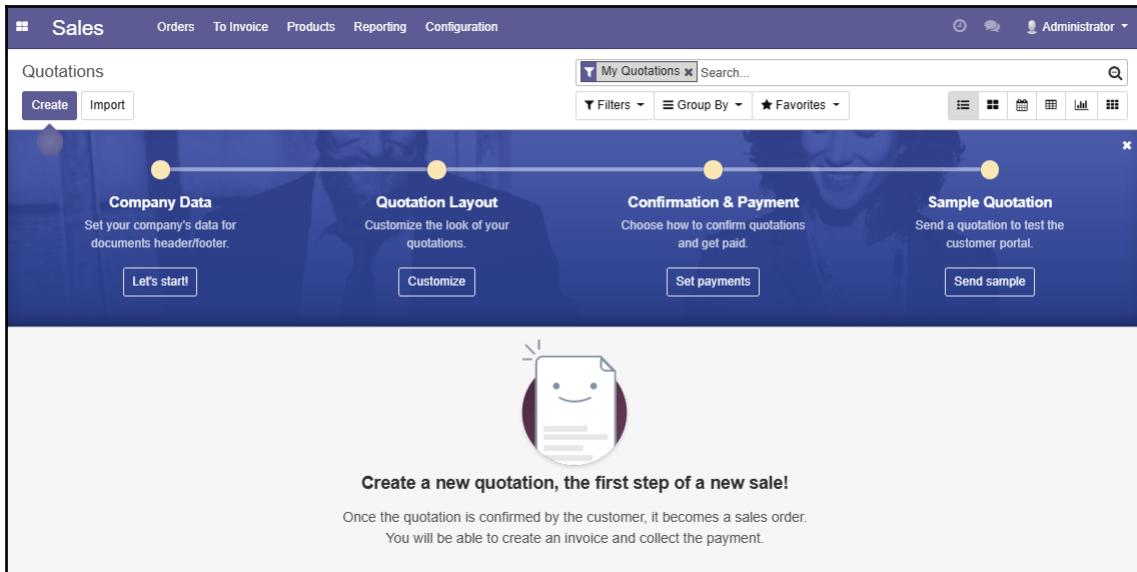
During the installation of modules and other long operations, you will often see an animated **Loading** icon at the center of your screen. Unlike previous versions of Odoo, which prompted you for accounting and other setup information, Odoo now completes the installation unattended.

Knowing the basic Odoo interface

After the installation of the sales order application, Odoo 12 takes you to the **Discuss** messaging app where your inbox and other communication activities are located. Notice the small purple teardrop that provides helpful tips.

Unlike previous versions of Odoo Community, which displayed menus across the top of the screen, Odoo 12 now nests applications in a menu on the far left. Click on **Sales** in the menu to bring up the sales application.

In Odoo 12, there is a brand new onboarding panel to configure the application:



This new onboarding panel makes it easier to identify the steps you need to take to successfully set up the sales application. Working from left to right, we can see that it starts with the **Company Data**. You then pick a **Quotation Layout** to customize the look of your quotations and sales orders. Next, you can define how you wish to set up your payments when a customer successfully completes an order. Finally, you can test your configuration by sending out a sample quotation through the customer portal.

Configuring company data

We will follow Odoo 12's advice here and begin configuring the **Sales** application by clicking on the **Let's Start** button in the **Company Data** section of the configuration panel. In this form, you can set the company data to what it needs to be for all of the respective fields:

Set your company data

Your logo Company Name

Silkworm Inc.

General Information

Address	102 Sezmore Dr.	Website	http://www.silkwormink.com
	Street 2...	Phone	
Murphysboro	Illinois (US) 62966	Email	
United States		Tax ID	
		Company Registry	
		Default incoterm	
		Currency	USD

Apply Cancel

Notice that both the **Company Name** and **Currency** fields are in purple, indicating that they are required fields. Fill in a name for your company and some address information, and then click **Apply** to save the company configuration data:

Sales Orders To Invoice Products Reporting Configuration

Quotations Create Import My Quotations Search... Filters Group By Favorites

Company Data Set your company's data for documents header/footer. All done! Customize

Quotation Layout Customize the look of your quotations. Customize

Confirmation & Payment Choose how to confirm quotations and get paid. Set payments

Sample Quotation Send a quotation to test the customer portal. Send sample

Create a new quotation, the first step of a new sale!

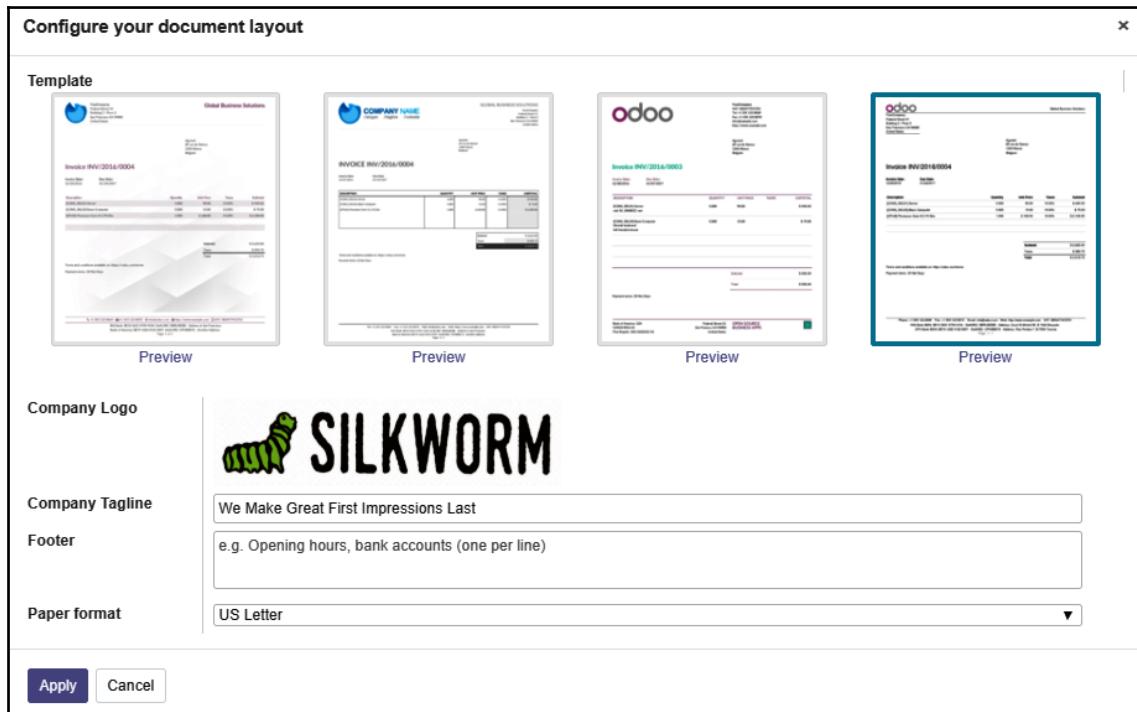
Once the quotation is confirmed by the customer, it becomes a sales order. You will be able to create an invoice and collect the payment.

The screenshot shows the Odoo Sales module. At the top, there are navigation links: Sales, Orders, To Invoice, Products, Reporting, and Configuration. Below this is a search bar with 'My Quotations' and a search input field. There are also buttons for 'Filters', 'Group By', and 'Favorites'. The main area is divided into four sections: 'Company Data' (with a note about setting company data for documents), 'Quotation Layout' (with a 'Customize' button), 'Confirmation & Payment' (with a 'Set payments' button), and 'Sample Quotation' (with a 'Send sample' button). A large central callout says 'Create a new quotation, the first step of a new sale!' and explains that once confirmed, it becomes a sales order. At the bottom, it says 'Once the quotation is confirmed by the customer, it becomes a sales order. You will be able to create an invoice and collect the payment.' A checkmark icon is visible next to the 'All done!' link under 'Company Data'."/>

You will see the **All Done** checkmark showing you that the company data has been saved.

Selecting your quotation layout

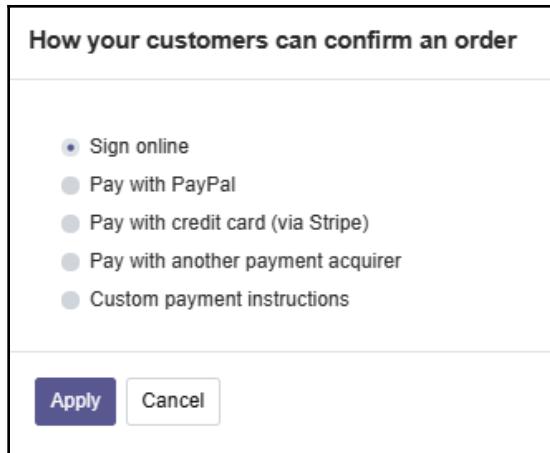
Another nice upgrade in Odoo 12 is the improved section of layouts available for quotations, sales orders, and invoices. This framework should make it easier to add additional layout options as well. The following screenshot is a preview of different layouts:



Click on **Customize** to select the quotation layout for your company.

Specifying payment options

Naturally, if you are in business, you are going to want to get paid. Odoo provides a variety of options to accept payment. We will explore some of those options in [Chapter 12, "Implementing E-Commerce with Odoo"](#), on eCommerce. For now, we will leave the default option, **Sign online**:



Select **Apply** to save the payment options.

Sending a sample quotation

Finally, you can send a sample quotation to test the operation on your system. Note that you will have to properly configure your email server to actually send it out. We will cover these options for configuration in Chapter 7, *Administering an Odoo Installation*. Once you complete the step and hit **Save**, you will get a confirmation that the initial configuration of the **Sales** application is complete:

The screenshot shows the Odoo Sales application interface. At the top, there are tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. The main area is titled "Quotations" with "Create" and "Import" buttons. The interface is divided into four sections with checkmark icons:

- Company Data**: Set your company's data for documents header/footer.
- Quotation Layout**: Customize the look of your quotations.
- Confirmation & Payment**: Choose how to confirm quotations and get paid.
- Sample Quotation**: Send a quotation to test the customer portal.

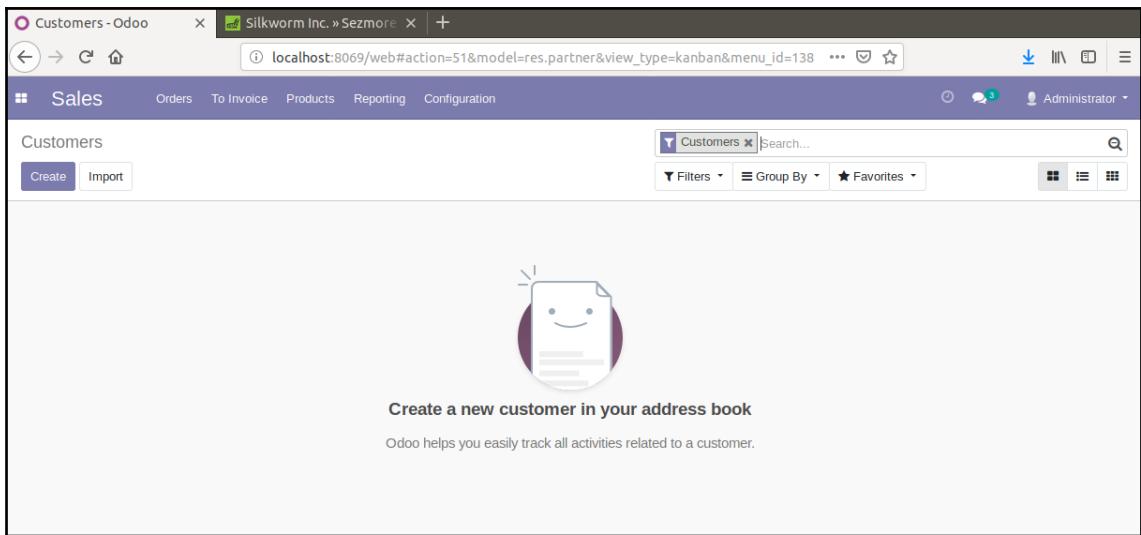
A message at the bottom says "Nice work! Your configuration is done." with a "Close" button. The main table displays the following data:

Quotation Number	Quotation Date	Customer	Salesperson	Total	Status
SO001	01/02/2019 17:59:12	Administrator	Administrator	\$ 1,414.50	Quotation Sent

After all the steps are completed and you click the **Close** button, the blue onboarding panel will disappear.

Creating your first customer

Now that we have finished setting up our **Sales** application, let's create a new customer. From the **Orders** menu at the top of the screen, choose **Customers**. Helpful instructions prompt you to begin entering your first customer into Odoo by clicking on the **Create** button:



This is the Odoo **Customers** form. Clicking **Create** will generate a customer record.

Silkworm sells t-shirts to both businesses and retail customers. For this example, we will use a fictional customer named **Mike Smith**, who wishes to purchase several t-shirts. Odoo offers flexibility in collecting customer information, and, by default, most fields are not required. If you see a field that is purple, that is a required field. In Odoo 12, the only required field for the customer is the name.

The rest of the fields are optional. Later, we will learn how you can configure Odoo to make the additional fields that are required. Consider this example:

The screenshot shows the Odoo Sales application interface. At the top, there's a navigation bar with tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the far right, it says "Administrator". Below the navigation bar, the title "Customers / New" is displayed, along with "Save" and "Discard" buttons.

The main form area starts with a section for selecting the customer type: "Individual" (radio button) and "Company" (radio button). The "Company" option is selected. The customer name "Mike Smith" is entered in the "Name" field, which has a blue background. Below the name is a dropdown menu set to "Company".

To the right of the name, there's a summary box with four metrics:

\$ 0 Sales	0.00 Invoiced	0 Analytic Acco...
0 Credit card(s)	Active	

Below the basic information, there are two columns of form fields:

Address	444 South Main	Job Position	e.g. Sales Director
	Street 2...	Phone	999-889-7777
	Marion	Mobile	
	Illinois (US)	Email	mikesmith@working_example.com
Tax ID	United States	Website	e.g. www.odoo.com
	e.g. BE0477472701	Title	Mister
		Language	English
		Tags	Walk In <input checked="" type="checkbox"/> Tags...

At the bottom of the form, there are several buttons: "Contacts & Addresses", "Internal Notes", "Sales & Purchases", "Invoicing", and an "Add" button.

In the preceding example, we have filled out some of the basic fields for our fictional customer, Mike Smith.

Specifying whether the customer is an individual or a company

At the very top of the form is a radio selection option to inform Odoo whether this customer is an individual or a company. For our example, we are using a walk-in retail customer. If you were doing a business-to-business type of operation, then often your customers would select the **Company** option.

Entering data into an Odoo form

Odoo utilizes a consistent interface to enter data throughout the application. Once you have learned how to enter data into one form, you should have no problem entering data into other forms in Odoo.

Required fields will always be in purple. If you see a purple field, you must fill in that data or you will not be able to save the record. You can move between fields by using your mouse or the *Tab* key. *Shift + Tab* will take you back to the previous field. Unlike some systems, you cannot move between fields in Odoo by using the arrow keys.

In many forms, you will have to select lists that allow you to choose from a list to populate the field. You can use your keyboard to type and limit the items that are displayed in a select list. By using the *Tab* key on your keyboard to find the appropriate item in the list, it is possible to enter data into a form with limited use of the mouse.



In the version of Odoo 12 installed for this chapter, it was necessary to first go down and choose the country before the states list can be populated. If you try to enter the state first, the list will be empty as there is no country selected.

Many select lists have two options at the bottom that will allow you to use additional search options or to create an item that is not in the list:

The screenshot shows the Odoo 'Customers / New' form. At the top, there are tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration, with Sales selected. Below the tabs, there are buttons for Save and Discard. The main form area has several sections: 'Address' (containing fields for Street, City, State, and Zip), 'Job Position' (with a dropdown for 'e.g. Sales Director' and a phone number '999-889-7777'), 'Mobile' (empty), 'Email' (mikesmith@working_exam...), 'Website' (e.g. www.odoo.com), 'Title' (Mister), 'Language' (English), and 'Tags' (with a 'Create and Edit...' button). On the left, there's a 'Tax ID' section with a dropdown showing Japanese prefectures: Hokkaidō, Aomori, Iwate, Miyagi, Akita, Yamagata, Niigata, Gifu, Toyama, Ishikawa, Fukui, Nagano, Gunma, Saitama, Chiba, Tokyo, Kanagawa, Hyogo, Kyoto, Osaka, Wakayama, Nara, Mie, Shiga, Gifu, Aichi, Miyazaki, Kumamoto, Oita, Fukuoka, and Kagoshima. At the bottom of the form, there are buttons for 'Contacts & Addresses' and 'Internal Notes', and a 'Send message' button at the very bottom.

In this example, we see a list of states with the option for additional searching or to create a new state that is not in the list.

For Language, Odoo has the ability to work with customers in a variety of languages. For our example, we will leave this as English. If, however, you were working with a company that preferred their documents in other languages, you could specify that language and Odoo will manage the translation.

Using multiple contacts and addresses

With Odoo 12, even an individual customer can have multiple contacts and addresses. For our example, we are not going to need multiple addresses. If you do need a new contact or address for your customer, then you can just click the **Add** button and a form will allow you to specify the contact and type of address.

Also, if you go one tab over, you can use the **Internal Notes** area to enter any additional notes that you wish to keep on the customer.

Editing customer Sales & Purchases

The bottom area of the customer screen is divided into a series of tabs or pages that assist in organizing customer information. In the **Sales & Purchases** page, we can assign options such as a salesperson and other sales-related options:

Contacts & Addresses	Internal Notes	Sales & Purchases	Accounting
Sale			
Is a Customer	<input checked="" type="checkbox"/>	Is a Vendor	<input type="checkbox"/>
Salesperson	<input type="text"/>		
Purchase			
Internal Reference	<input type="text"/>	0 Bank account(s)	0 Credit card(s)
Misc			
Payments			

The available options in the customer **Sales & Purchases** page are as follows:

- **Is a Customer:** This is the customer checkbox known in Odoo as a **Boolean** field. It can be marked checked or unchecked. Odoo has a unique method of storing data related to people in the system. All individuals are stored in the same table (`res_partner`) regardless of whether they are a customer or supplier. The customer flag tells Odoo that this is, in fact, a customer record. This field must be checked for Odoo to recognize Mike Smith as a customer.
- **Salesperson:** The salesperson field allows you to select who the direct salesperson will be for this customer. While the field is not required, it is often populated if you are integrating your sales management system with the **Customer Relationship Management (CRM)** module. We will use this field in Chapter 3, *Exploring Customer Relationship Management in Odoo 12*; for now, we can leave the field blank.
- **Internal Reference:** Often, when implementing Odoo, a company already has an existing customer numbering system in place. The reference field is the perfect field to populate with an existing customer number. Otherwise, this field can be left blank or used for another purpose. For our example, we are going to leave this field blank.
- **Is a Vendor:** Because Odoo stores customer and vendor data in the same table, it is possible to be both a customer and a supplier. In this example, we will keep Mike Smith as a customer only.



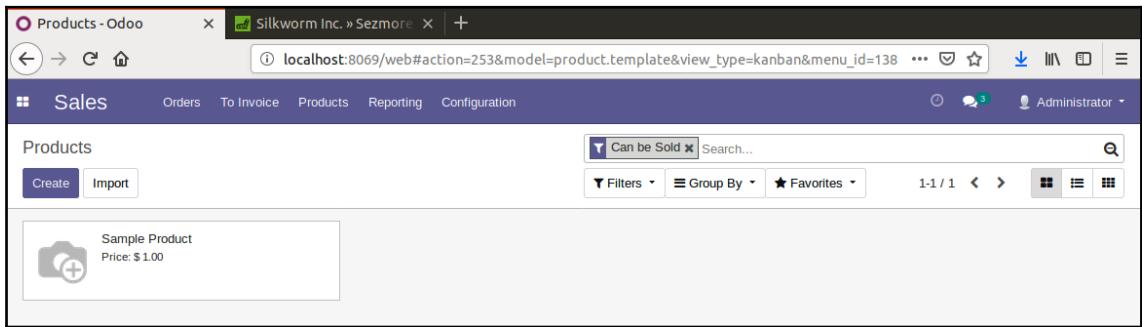
Odoo uses a common database to store customer and supplier records. This makes it easier to manage data as customers and suppliers are designated by simple checkboxes in the **Sales & Purchases** page of the customer screen.

Editing customer accounting

The **Accounting** tab has been simplified in Odoo 12 to simply contain a list of bank accounts associated with a customer. As this is a walk-in customer, we have no need to collect this information.

Entering a product in Odoo

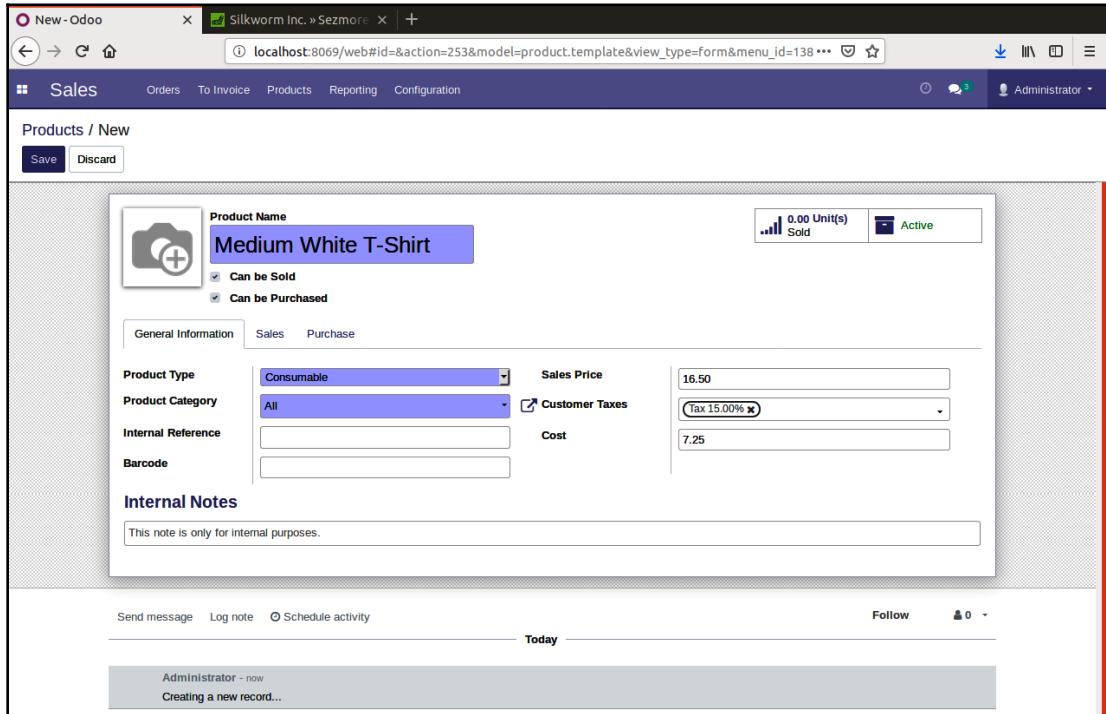
Now that we have a customer, it is time we enter some products to sell to our new customer. For our example, we are going to enter a medium white cotton t-shirt. Click on the **Products** menu at the top of the page and choose the **Products** menu to pull up the screen:



Now, in Odoo 12, even if you specify that you do not want demonstration data, you still get a **Sample Product** for **\$1.00**. Let's ignore this product and create a new product for our t-shirt.

Creating products in Odoo

Create a new product by clicking on the **Create** button:



The preceding screenshot is for the **General Information** page of the product form for entering a product record into Odoo.

Product name

The **Product Name** is what will be displayed on the sales orders and invoices and in all other screens that refer to this specific product. For our example, we are selling **Medium White T-Shirt**.

Can be sold

Much like the customer active flag, you can use **Can be Sold** to remove products from showing up on product lists by unchecking **Can be Sold**. For our example, we want to sell this t-shirt to **Mike Smith**, so we will leave the option checked.

Can be purchased

Even though we have not yet installed the purchasing system, Odoo 12 lets you specify whether a product can be purchased. We will accept the default that, in addition to selling this t-shirt, we can purchase them as well. This will play an important function when we get to Chapter 4, *Purchasing with Odoo*.

Product type

Product Type is the first option in the **Information** page on the product screen. There are two available product types:

- **Consumable**
- **Service**

Service product types will not create procurement in purchase orders. Consumables are products that you actually sell and can be configured to generate purchase orders. For our example, we will set the product type to **Consumable**.

Internal reference

For the most part, Odoo utilizes the **Name** field and the description when displaying product information. It is very common that a company may have a coding system for their products. The **Internal Reference** field is useful to enter an alternative product code or number for the product. In this example, we will leave the **Internal Reference** field blank.

Sale price

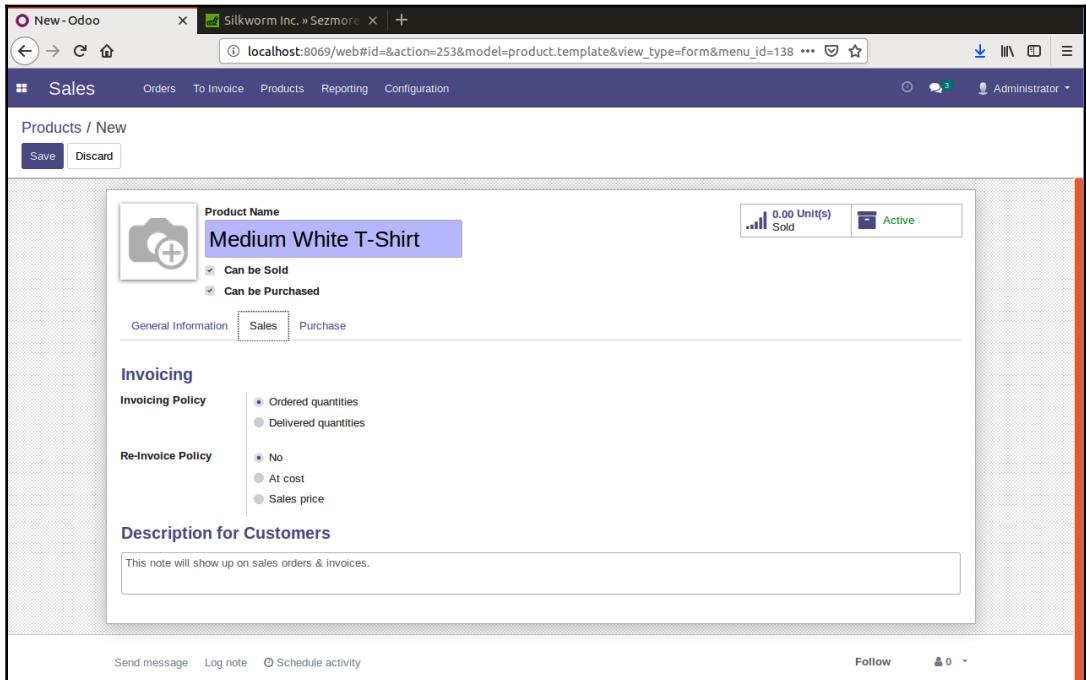
This field sets the sales price of the item as it will appear on the sales order. For our example, we are setting the **Sales Price** of the t-shirt to \$16.50.

The Cost price

This field sets the cost of the item that can be utilized for simple profit margin calculations. We will set a **Cost** of \$7.25 for the t-shirt.

Checking out the Sales tab

In Odoo 11, the **Sales** tab was conspicuously left blank by default. Now, in Odoo 12, you can use it to specify the invoicing policy and the reinvoicing policy, as shown in the following screenshot:



We also have a specific **Description For Customers** for the product that will be displayed to the customer. This can be a different description to how you refer to the product internally and can be a much longer description.

Entering a product Invoicing page

By default, Odoo has set up a tax of 15% for both our **Customer Taxes** and our **Vendor Taxes**. However, there will be times when you have a product that has a specific tax. In the United States, one example is that cigarettes often have a more substantial tax than other items such as food. Odoo allows you to specify additional tax options for a given product on the **Accounting** page. Taxes can be specified for both the customer and the supplier separately.

The following screenshot is of the **Accounting** page located in the **Product** form:

The screenshot shows the 'Invoicing' tab selected in the Odoo Product form. It displays settings for receivables and payables taxes, and the invoicing policy.

Section	Setting	Value
Receivables	Customer Taxes	Tax 15.00%
	Payables	Vendor Taxes
Invoicing	Invoicing Policy	<input checked="" type="radio"/> Ordered quantities <input type="radio"/> Delivered quantities

You will also notice that Odoo allows you to have multiple taxes for the same product. This would allow you to have a base tax that would apply to all the products and then simply add an additional tax, or even a tax credit, depending on that specific product.

Invoicing policy

By default, Odoo configures invoicing so that the line items of the invoice will be created depending on the ordered quantities from the sales order. This means that, even if none of the items have been shipped, the customer will still be invoiced. Alternatively, you can change the invoicing policy so that the customer is invoiced on delivered items. If you have items on a sales order that have not been shipped yet, the customer will not be invoiced for those items.

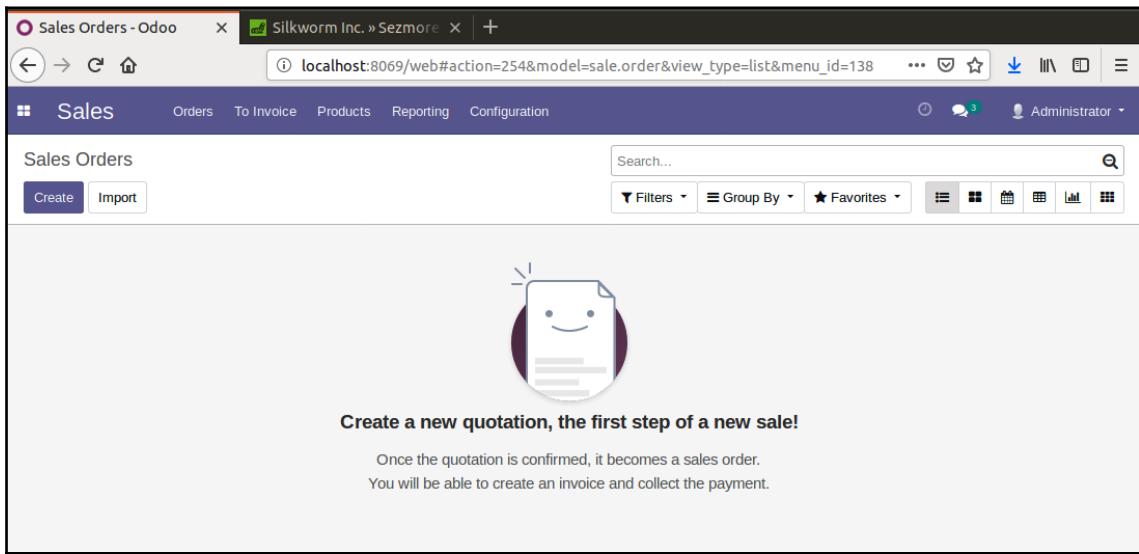
Saving the product

Clicking the **Save** button stores the product record in Odoo. If you click on **Discard**, you will be prompted with a warning message telling you that you will lose your changes.

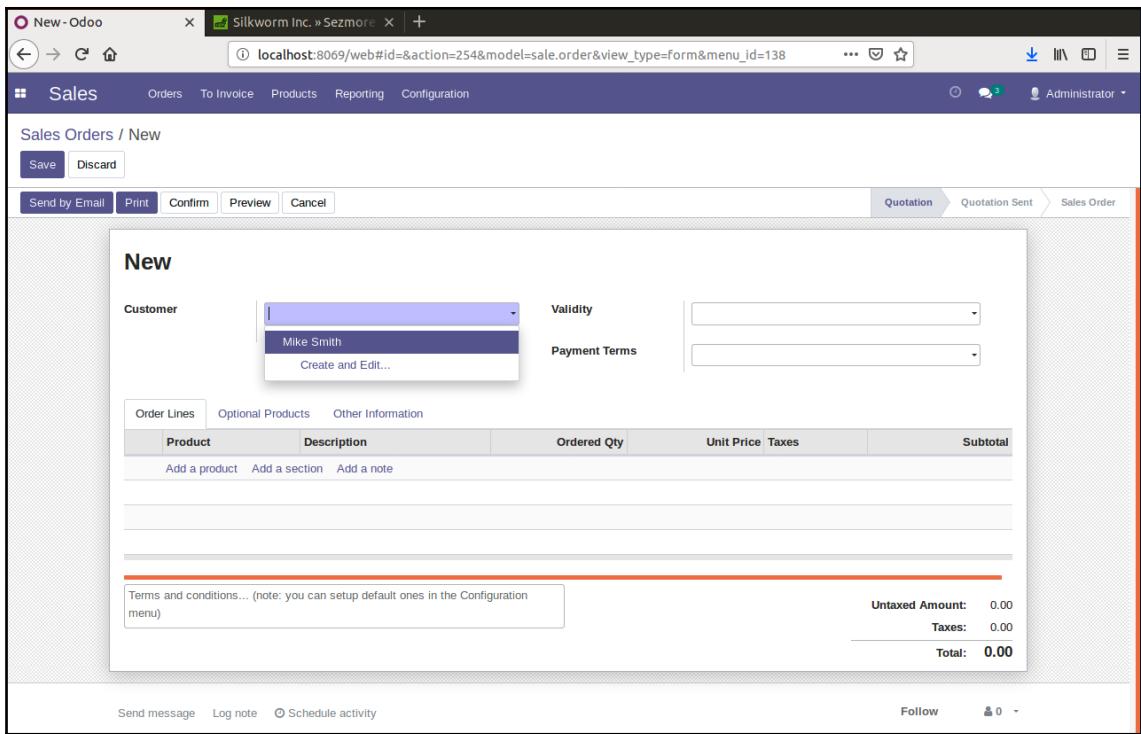
Entering your first sales order

Now for the moment we have all been waiting for. We finally get to sell our products by entering a sales order. To get to the **Sales Orders** screen, click **Orders** in the top-left menu and then choose the **Sales Orders** menu item.

The following screenshot shows existing **Sales Orders** and allows users to create a new sales order:



Click on the **Create** button to create a new sales order. Every brand-new sales order begins as a quotation and stays in that state until you confirm the sale. Only after confirming the quotation will your sale be referred to as a sales order. Let's have a look at the following screenshot:



The preceding screenshot is of a new sales order form with the cursor set on the **Customer** field.

Selecting the customer

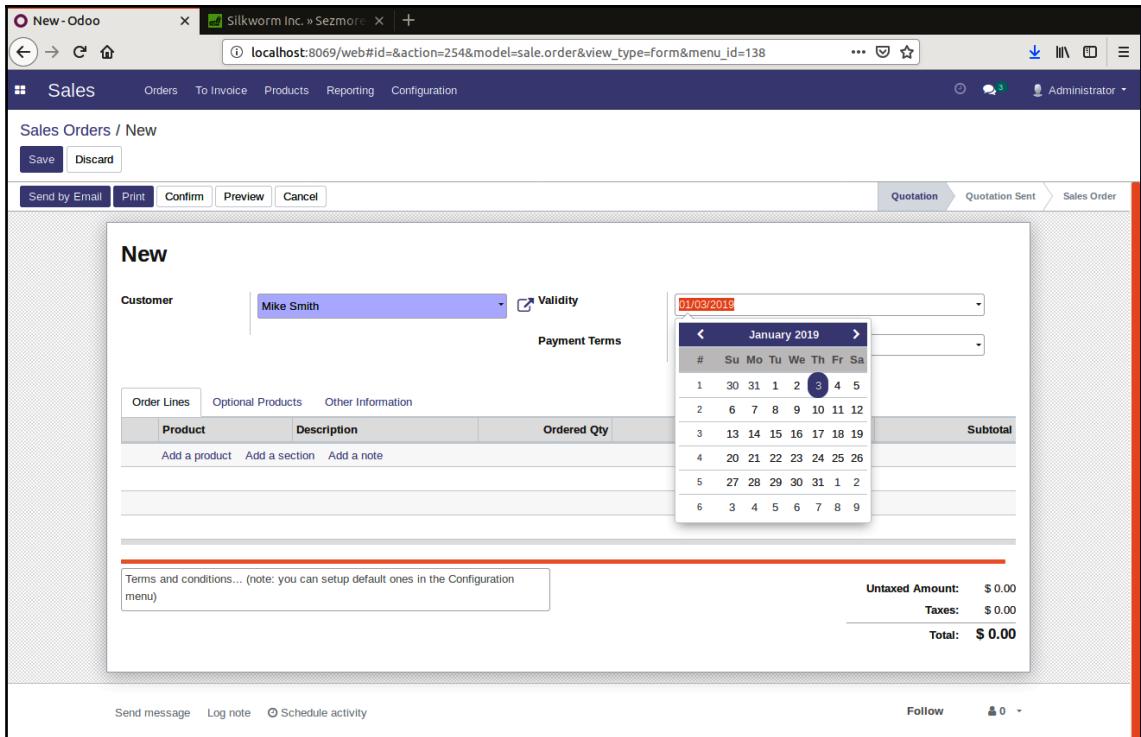
When you create a new quotation sales order, you are prompted to first select the customer from the select list. As you add customers, you will have the option to search and locate customers for the sales order. For now, we will be selecting the customer we entered earlier on in this chapter, **Mike Smith**.



You will not be able to begin entering line items until you have specified the **Customer** for the sales order.

The Validity date

By default, there is no expiration date for the quote. However, if you would like to specify a date in which the quote will no longer be valid, you can specify it here:



For date fields, you may type the date or use the little pop-up calendar to select the proper date. Since this field is not required, you may also clear its value by pressing the *Delete* or the Backspace key while the date is selected.

Payment terms

Odoo will automatically bring in the payment terms for the customer you select. On the sales order, you have the option to override the customer's payment terms for the specific sales order.

Entering line items on a quotation sales order

Now, we are ready to begin specifying the **Product** we wish to sell. Click on **Add a product** in the line item area to add a line to the grid. The first field will be **Product**. Select **Medium White T-Shirt** from the list box:

The screenshot shows the Odoo Sales Orders / New interface. At the top, there are tabs for Quotation, Quotation Sent, and Sales Order. Below the tabs, there are buttons for Save, Discard, Send by Email, Print, Confirm, Preview, and Cancel. The main area is titled "New" and has sections for Customer (Mike Smith), Validity (01/03/2019), and Payment Terms. The "Order Lines" tab is selected, showing a table with columns: Product, Description, Ordered Qty, Unit Price, Taxes, and Subtotal. A single line item is listed: Medium White T-Shirt, Medium White T-Shirt, 1.000, 16.50, Tax 15.00%, and \$ 16.50. Below the table are links for Add a product, Add a section, and Add a note. At the bottom, there is a note about terms and conditions, and a summary table showing Untaxed Amount (\$ 16.50), Taxes (\$ 2.48), and Total (\$ 18.98). At the very bottom, there are buttons for Send message, Log note, and Schedule activity, along with a Follow button and a notification count of 0.

The fields of your **Order Lines** should populate and look similar to what's shown in the preceding screenshot.

The Product field

Each line item starts out by selecting the product. You can add products on the fly by choosing **Create and Edit...** from the bottom of the list. Once there are more products in the list, you can also bring up a product search window using the **Search More...** option. After you select the **Product** field, Odoo retrieves the tax and pricing information from the server to display in the line item.

Description

Odoo will pull over the **Description** from the **Product** record to populate the **Description** field on the line item. It is possible to override the **Description** on the quotation sales order. For this example, we will leave the **Description** as it's pulled over from the **Product** record.

Ordered quantity

The product quantity will be 1 as default. Naturally, you will change this field to the quantity of products you have sold. We will just leave the quantity as 1 for this example.

Taxes

Odoo supports taxes by line item and will automatically pull over the 15% tax rate that we have defined in the **Product** record. Additional taxes can be added or removed from the line item. For this example, we will leave the tax at 15%.

Unit price

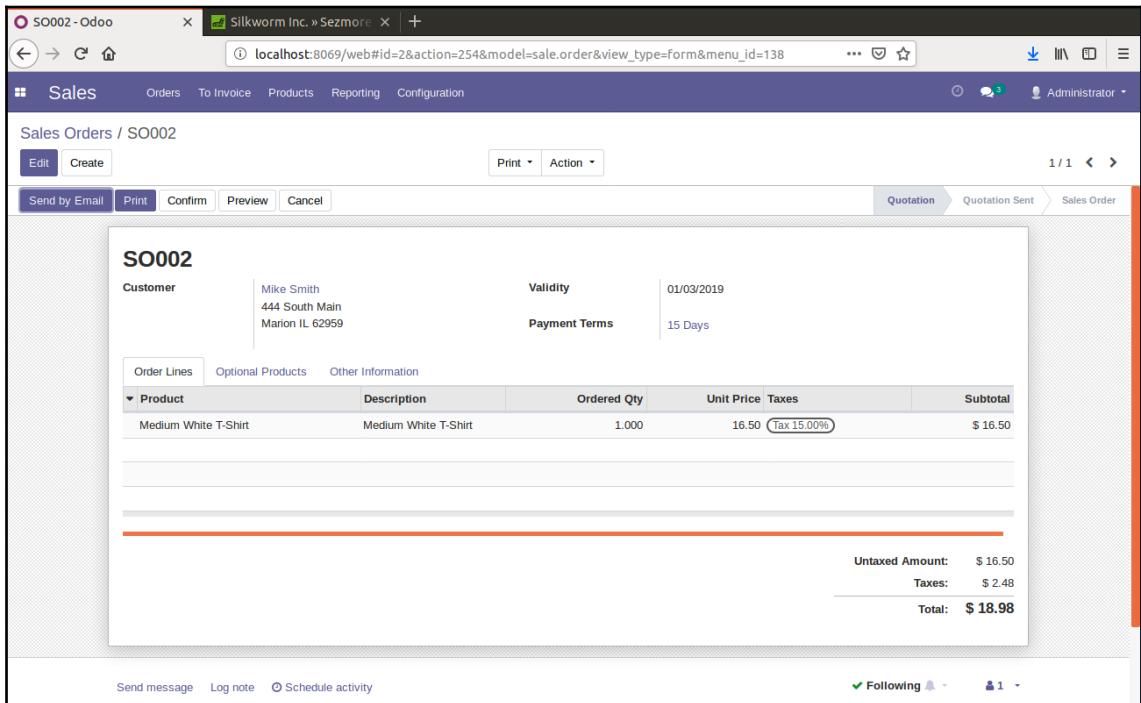
Odoo pulls the sale price from the product record to populate the unit price in the line item. It is possible to override the price in the line item. For this example, we will leave the unit price at \$16.50.



Be careful of changing prices in the line items of Odoo. It is possible that, if you click back on the **Product** field or tab through other fields in the line item, the unit price will flip back the price in the **Product** record. If you are changing prices in the line items, make sure to double-check your unit prices before you confirm your sales order.

Saving the quotation sales order

Click **Save** to store the quotation. The form will refresh, displaying the full customer address as well as updating the tax and final total of the quotation sales order:



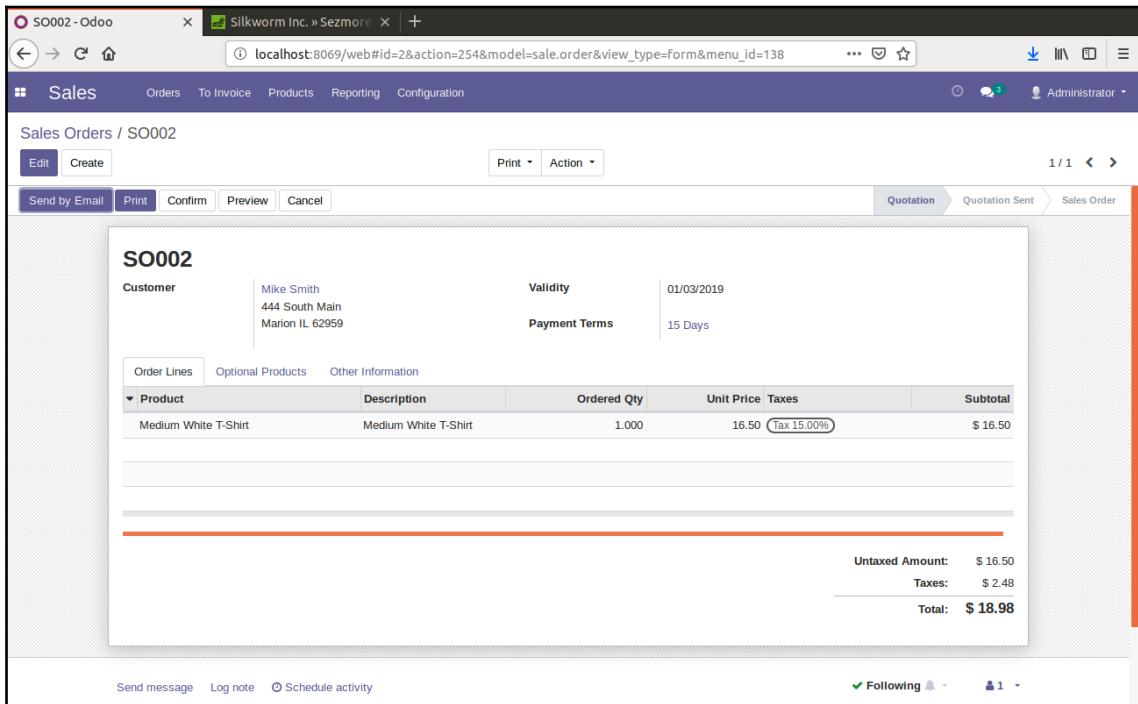
The preceding screenshot is of our first quotation in Odoo.

Understanding the sales order workflow

Although we started out entering a sales order, the current state of this order is a **Quotation**. Odoo 12 displays the current state of transactions in the top-right corner of the form.

This indicator makes it very easy to see the current stage of a transaction throughout the Odoo workflow. In this example, we can see that this is currently a **Draft Quotation**. We can also see that the quotation would first need to be sent and/or be confirmed as a sales order, before the quotation can finally be considered **Done**.

The available actions you can take on this quotation are displayed in the top-left corner of the form:



The preceding screenshot is of available actions for an Odoo quotation.

Send by Email

Clicking the **Send by Email** button will send a copy of the quotation to the email address in the customer's file. Setting up your email configuration will be a topic for another time (Chapter 7, *Administering an Odoo Installation*).

Print

Even in the digital age, it is still very common to need a printed copy of a quotation or sales order. Clicking on the **Print** button will generate a PDF document containing your quotation.

Confirm Sale

The **Confirm Sale** button will convert the quotation into a sales order and push the transaction further down the sales workflow.

Cancel

Clicking the **Cancel** button will prompt you to cancel this quotation. The quotation is not deleted and can still be viewed. Canceling the quotation ends the sales order workflow, and the quotation will only be kept in the system for archive purposes.

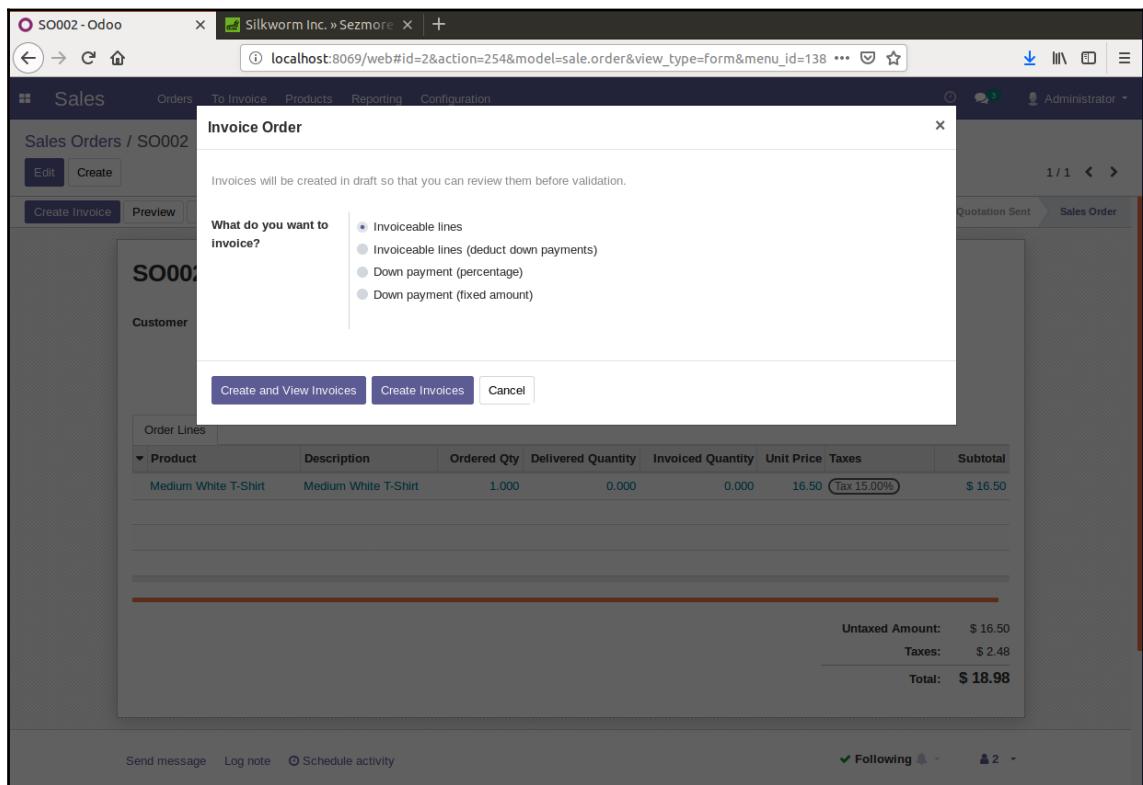
Click on the **Confirm Sale** button to convert this quotation into a sales order:

The screenshot shows the Odoo Sales module interface. The top navigation bar includes tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. The main title is "Sales Orders / SO002". Below the title are buttons for Edit, Create, Print, Preview, Send by Email, Cancel, and Lock. A status indicator shows "Quotation" on the left and "Sales Order" on the right. The main content area displays the sales order details for SO002. It includes sections for Customer (Mike Smith, 444 South Main, Marion IL 62959), Confirmation Date (01/02/2019 18:35:09), and Payment Terms (15 Days). The Order Lines section lists one item: Medium White T-Shirt, Description: Medium White T-Shirt, Ordered Qty: 1.000, Delivered Quantity: 0.000, Invoiced Quantity: 0.000, Unit Price: 16.50, Taxes: Tax 15.00%, and Subtotal: \$ 16.50. At the bottom, financial summary lines show Untaxed Amount: \$ 16.50, Taxes: \$ 2.48, and Total: \$ 18.98. Navigation icons at the bottom include Send message, Log note, Schedule activity, Following, and a user count of 2.

You will see the status change for the sales order from **Quotation** to **Sales Order**.

Invoicing the sale

Depending on the workflow of the business, a lot of different things can happen after you have confirmed a sales order. In manufacturing companies, you may need to both purchase products and create a manufacturing order to produce the final product before you invoice the customer. In our example, we are going to go ahead and invoice the customer for the t-shirt they have ordered. Click on the **Create Invoice** button to generate an invoice for the sales order. An Odoo **Invoice Order** wizard pops up to walk you through the invoice creation process:



The preceding screenshot is of the **Invoice Order** wizard.

What do you want to invoice?

Odoo provides a variety of options for invoicing the entire sales order or instead invoicing based on other methods. The available choices are as follows:

- **Invoiceable lines (deduct down payments):** Choose this option if you want to invoice the lines and deduct any down payments you have received. This is the default option. Using the invoiceable lines without deducting down payments will simply ignore any of those payments when producing the invoice.
- **Down payment (percentage):** You will be prompted to enter the percentage amount of the down payment.
- **Down payment (fixed amount):** You will be prompted to enter a fixed amount of the down payment.

Creating the invoice

For our example, we will be using the default option. As we have no down payments, Odoo will process the sales order in the same way as if you chose the first option, **Invoiceable lines**. Click **Create Invoices and View Invoices** to generate the invoice. Initially, the invoice is created in a **Draft** state. Clicking **Validate** will confirm the invoice and post the transaction.

If you have followed along and everything worked as it should, then you will see an invoice similar to the following one:

The screenshot shows the Odoo web interface for creating an invoice. The top navigation bar includes links for Sales, Orders, To Invoice, Products, Reporting, Configuration, and a user dropdown for Administrator. The main title is "Sales Orders / SO002 / Invoice". Below the title, there are buttons for Edit, Create, Print, Action, Validate (which is currently selected), and Draft/Open/Paid status indicators. The invoice details section shows the following information:

Customer		Mike Smith 444 South Main Marion IL 62959	Invoice Date	
			Due Date	
Salesperson		Administrator		
Sales Team		Sales		

Below this, the "Invoice Lines" tab is selected, showing a single line item:

Product	Description	Quantity	Price	Taxes	Subtotal
Medium White T-Shirt	Medium White T-Shirt	1.000	16.50	(Tax 15.00%)	\$ 16.50

At the bottom right, the total amounts are displayed:

Untaxed Amount:	\$ 16.50
Tax:	\$ 2.48
Total:	\$ 18.98

At the very bottom of the page, there are links for Send message, Log note, Schedule activity, and a Following indicator with 1 item.

At this time, it is worth noting Odoo's use of an interface feature called **breadcrumbs**. These links, which appear on form views just below the topmost menu, allow you to traverse from your invoice back to the relevant sales order from which it was derived. The use of these links is the preferred method of backtracking to prior screens, as opposed to using your browser's back button.

Summary

In this chapter, we started by creating an Odoo database. We then installed the Sales Order Management module and created our first customer. With our customer created, we turned our attention to setting up a product in Odoo and entering our basic company information. Next, we created a quotation and followed the workflow all the way through to confirming the sales order and generating an invoice. In the next chapter, we'll look at our sales strategy and what we want to achieve via CRM software.

3

Exploring Customer Relationship Management in Odoo 12

Until recently, most business and financial systems had product-focused designs, while records and fields maintained basic customer information, processes, and reporting, typically revolving around product-related transactions. In the past, businesses were centered on specific products, but now the focus has shifted to center the businesses on customers. A **Customer Relationship Management (CRM)** system provides the tools and reporting necessary to manage customer information and interactions.

In this chapter, we will cover the following topics:

- Exploring what it takes to implement a CRM system as part of an overall business strategy
- Installing the CRM application and setting up salespersons that can be assigned to our customers
- Learning how to create and manage leads
- Creating opportunities and schedule events in Odoo
- Discovering Odoo's Chatter feature
- Managing multiple sales teams

Using CRM as a business strategy

Before jumping into the specific CRM features of Odoo, it is valuable to briefly discuss the importance of a comprehensive approach to implementing a CRM system in your business. The fact is that successfully implementing a CRM system requires much more planning than just installing software and asking employees to fill in the data. CRM software systems are only a technical tool for assisting your sales and marketing department in acquiring and keeping customers. Certainly, software will play an important role, but to obtain real benefits from a CRM system, you must first do extensive research to understand your customer and exactly how you wish to shape their customer experience.

It is critical that the sales people share account knowledge and completely understand the features and capabilities of the system. They often have existing tools that they have relied on for many years. Without clear objectives and goals for the entire sales team, it is likely that they will not use the tool. A plan must be implemented to spend time training and encouraging the sharing of knowledge to successfully implement a CRM system.

Managing customer experience

Today, customers face a wide range of choices when it comes to purchasing products and services. At the most fundamental level, customers often develop great loyalty to brands with which they have a positive customer experience. Companies such as Apple and Harley Davidson are successful largely because of fierce brand loyalty based on positive customer experiences. Making the most of a CRM system requires you to put yourself in the role of your customer and develop a consistent strategy to improve their overall customer experience.

Treating your customer like a real person

As computers became more common, it wasn't long until people began to feel as though they were being treated like a number by many companies. In many ways, CRM systems turn the tables around. Instead of treating customers like cattle, a smart account manager using a CRM system can greatly personalize the customer experience. Treat your customer like an individual and they will reward you with their loyalty.

Because you are looking to create a very personalized customer experience, it is important to thoroughly look at your customers' interactions with the company when designing your own CRM system. A company who sells high-end security systems to government institutions will need to provide drastically different customer experiences than a company that is marketing a pool maintenance service to homeowners.

Using your mission statement to drive the design of your CRM system

A good CRM system will be built around the core goals and mission of your company. If your company does not have customer-focused goals, then you should address that before beginning to design a CRM system. More importantly, there needs to be a focus on any concerns and interactions that have a direct impact on the customer experience. A good CRM system will not just manage the sales process, but the entire customer experience and interactions before, during, and after the sale.

A real-world case study – improving customer experience

Now, we will take a detailed real-world look at how a CRM system can be implemented to improve the customer experience. We'll begin by looking at the following company slogan:

We make great first impressions last

Here, we have a slogan that certainly speaks of the value of the customer experience. To make that great first impression and keep it, there are several critical service expectations to consider:

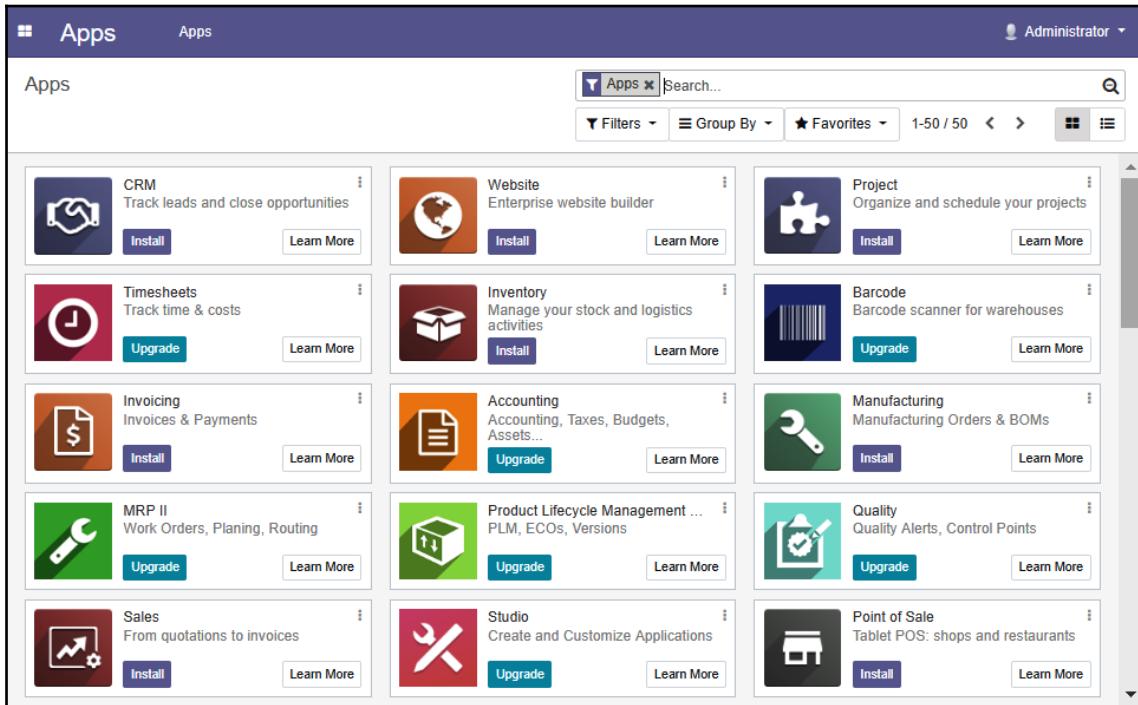
- Orders must be accurate and easy for customers to place.
- Orders must be delivered on time.
- The quality of customer service must be excellent.

While listing these customer service goals may seem obvious, explicitly naming your objectives is important when building a CRM system. This is because there is a natural tendency, when building a CRM system, to focus almost exclusively on customer acquisition and presale activities. Therefore, we must take care to remember that a CRM system must also support processes that manage the entire customer experience. Here are the types of questions that you should consider when building your own CRM system:

- How are problem orders handled?
- How is the customer contacted if there is a product back order?
- If the customer calls, can the service representative easily provide delivery tracking information?

Installing the CRM application

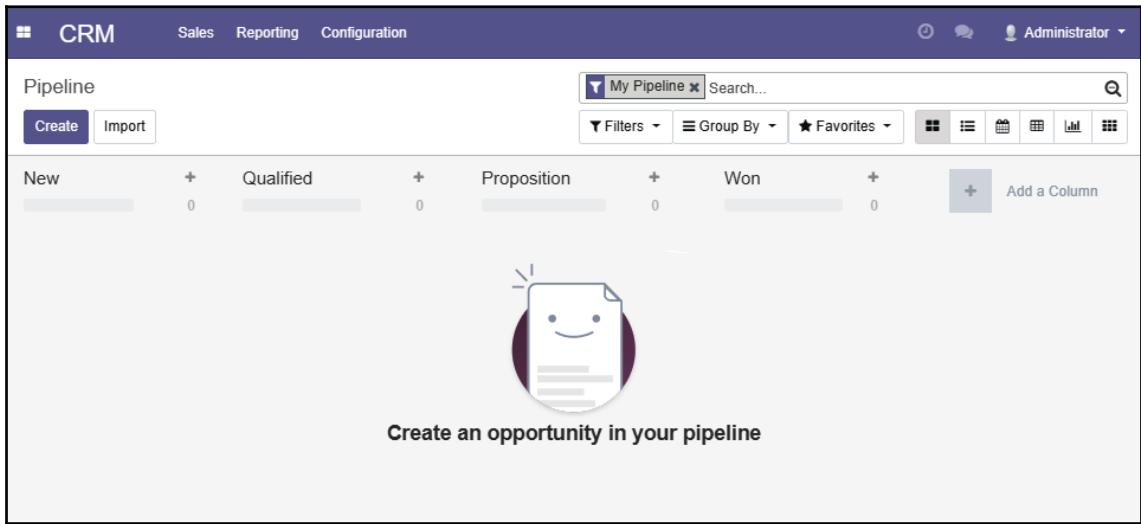
If you have not installed the CRM module, log in as the administrator and then click on the **Apps** menu. In a few seconds, the list of available apps will appear. **CRM** will likely be in the top-left corner of the screen:



Click on **Install** to set up the CRM application.

Your first look at the Odoo CRM pipeline

Just as with the installation of the Sales application, Odoo takes you to the **Discuss** menu. Choose **CRM** from the app menu in the top-left-hand side. The following screenshot indicates what the CRM dashboard looks like before any opportunities have been created:



In Odoo 12, you are presented with a blank **Pipeline** that will serve as the primary way to track your opportunities.

Assigning the sales representative or account manager

As in most CRM systems, the sales representative, or account manager, plays an important role in Odoo 12. Typically, this is the person that will ultimately be responsible for the customer account and a satisfactory customer experience.



Often, a company will use real people as their salespeople; however, it is certainly possible to have a salesperson record refer to a group, or even a subcontracted support service instead.

We will begin by creating a salesperson that will handle standard customer accounts. Note that a sales representative is also a user in the Odoo system.

Create a new salesperson by going to the **Settings** menu, selecting **Users & Companies**, then **Users**, and, finally, clicking on the **Create** button. The new user form will appear. We have filled in the form with values for a fictional salesperson, Terry Zeigler:

The screenshot shows the Odoo user creation interface. At the top, there's a header bar with tabs for Settings, Dashboard, Users & Companies, Translations, General Settings, and an Administrator dropdown. Below the header, the main form has a title 'Users / New'. It includes two buttons: 'Save' and 'Discard'. A status bar at the top right shows 'Never Connected' and 'Confirmed'. The main form fields are: a placeholder image with a camera icon, a Name field containing 'Terry Zeigler' with a checked 'Active' checkbox, an Email Address field containing 'terryzeigler@exampleemail.com', and tabs for 'Access Rights' and 'Preferences'. Under 'Application Accesses', there are dropdown menus for Sales (set to Manager), Accounting & Finance (set to Billing Manager), and Administration. In the 'Other' section, there's a checkbox for 'Access to Private Addresses' which is unchecked. The entire interface is styled with a light blue and white color scheme.

The preceding screenshot is of the user's **Access Rights** tab.

Specifying the name of the user

You specify the username in the same way that you assigned the name of your customer in the preceding chapter. Unlike some systems that provide separate first name and last name fields, with Odoo you specify the full name within a single field.

Email address

Beginning in Odoo 9, the user and login forms prompt for your email as opposed to your username. This practice has continued through Odoo version 12 as well. It is still possible to use a username instead of an email address, but given the strong encouragement to use an email, it is possible that, in future versions of Odoo, the requirement to provide an email address may be more strictly enforced.

Access Rights

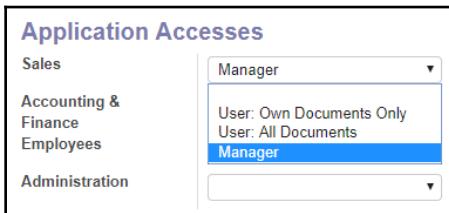
The **Access Rights** tab allows you to control which applications the user will be able to gain access to.



Depending on the applications you have already installed, or the dependencies Odoo may add in various releases, it is possible that you will have other **Access Rights** listed.

Sales application settings

When setting up your salespeople in Odoo 12, you have three different options to choose from, depending on how much access an individual user has to the sales system, as you can see in **Application Accesses**:



By choosing **Manager**, the user is granted the highest level of access within the **Sales** application. We will go through each of these options in the following subsections.

User – own documents only

This option offers the most restrictive access to the Sales application. Any user with this level of access is only allowed to see the documents they have entered themselves or those that have been assigned to them. They will not be able to see any leads that have been assigned to other salespeople in the Sales application.

User – all documents

With this setting, a user will have full read and write access to all documents within the Sales application.

Manager

The **Manager** setting is the highest access level in the Odoo sales system. With this level of access, a user can see all the leads and access the configuration options of the Sales application. The **Manager** setting also allows the user to access statistical reports.

We will leave the other **Access Rights** options unchecked. These are used when working with multiple companies or with multiple currencies.

Now, let's take a look at the **Preferences** tab for users:

The screenshot shows the Odoo User Preferences interface for a user named Terry Zeigler. At the top, there is a profile picture placeholder, the name "Terry Zeigler", an email address "terryzeigler@examplemail.com", and a status box indicating "Active" with a checkmark. Below this, there are two tabs: "Access Rights" (which is currently selected) and "Preferences". The "Access Rights" tab displays the "User Groups" section with "Sales Manager" checked and "Sales User" unchecked. The "Preferences" tab displays the following sections: "Localization" (Language: English, Timezone: America/Chicago), "Messaging and Social" (Notification Management: Handle by Emails, Alias: Best Regards, Signature: Terry Zeigler, Sales Manager), and "Advanced" (Signature: Best Regards, Terry Zeigler, Sales Manager). A note at the bottom states: "This user has been granted the Sales Manager role, which includes access to the Sales module and its features." A "Save" button is located at the bottom right.

The **Preferences** tab consists of the following options.

Language and timezone

Odoo allows you to select the language and timezone for each user. Currently, Odoo supports more than 20 language translations. Specifying the **Timezone** field allows Odoo to coordinate the way that the date and time is displayed in messages.



Leaving **Timezone** blank for a user can sometimes lead to unpredictable behavior in the Odoo software. Make sure that you specify a **Timezone** when creating a user record. Better yet, check [Chapter 15, Discovering Custom Odoo Modules](#), on customizing Odoo, to learn how you can make **Timezone** a required field!

Messaging and social – notification management

In Odoo 12, you have the choice of having your notifications sent to your email inbox or having your notifications sent to the inbox in Odoo. In this way, each user can work with Odoo in a way that best fits their personal preferences.

Signature

The **Signature** section allows you to customize the text that will automatically be appended to the bottom of all Odoo-generated messages and emails sent by this user.

Manually setting the user password

You may have noticed that there is no visible password field in the user record. This is because the default method is to email the user an account verification that they can use to set their own password. However, if you do not have an email server configured, there is an alternative method for setting the user password.

After saving the user record, use the **Change Password** function from the **Action** menu. A form will then appear that allows you to set the password for the user:

The screenshot shows a modal dialog titled "Change Password". It has two input fields: "User Login" containing "terryzeigler@examplemail.com" and "New Password" containing "....". Below the fields are two buttons: "Change Password" (in blue) and "Cancel".

Just enter the **New Password** and click on **Change Password**.

Assigning a salesperson to a customer

Now that we have set up our salesperson, it is time to assign the salesperson their first customer. Previously, no salesperson had been assigned to our one and only customer, Mike Smith. So, let's go to the **CRM** menu, click on the **Customers** menu option under **Sales**, and then click on **Mike Smith** to pull up his customer record, edit it, and assign Terry Ziegler as his salesperson. The following screenshot shows the **Customer** screen with the **Sales & Purchases** tab open in order to assign a **Salesperson**:

The screenshot shows the "Customer" screen in Odoo. The "Sales" tab is active, displaying the "Sale" section. Under "Sale", there is a "Salesperson" field with a dropdown menu open, showing "Terry Zeigler" selected. Other options in the dropdown include "Administrator" and "Create and Edit...". To the right of the dropdown is a "Purchase" section with a "Is a Vendor" checkbox. Below the "Sale" section is a "Misc" section with an "Internal Reference" field. On the right side of the screen are "Payments" sections showing "0 Bank account(s)" and "0 Credit card(s)".

Here, we have set the salesperson to **Terry Zeigler**. By assigning your customers a **Salesperson**, you can then better organize your customers for reports and additional statistical analysis.

Understanding your pipeline

You use the pipeline to organize your opportunities according to what stage they are at within your sales process. Click on **Pipeline** in the **CRM** menu to see the overall layout of the **Pipeline** screen, as shown in the following screenshot:

The screenshot shows the Odoo CRM Pipeline screen. At the top, there are tabs for CRM, Sales, Reporting, and Configuration, along with an Administrator dropdown. Below the tabs, the word "Pipeline" is displayed, followed by a search bar with the placeholder "My Pipeline" and a magnifying glass icon. To the right of the search bar are buttons for "Filters", "Group By", and "Favorites". Further right are icons for different view modes: grid, list, calendar, and chart. Below these controls, there are four stages represented by horizontal bars: "New" (0), "Qualified" (0), "Proposition" (0), and "Won" (0). Each stage has a plus sign (+) to its left and a plus sign (+) to its right, indicating the ability to add more stages. At the bottom of the pipeline area, there is a large circular icon featuring a smiling face and the text "Create an opportunity in your pipeline".

In the preceding **Pipeline** form, one of the first things that you may notice is that there are default filters applied to the view. In the top search box, you will see that there is a filter to limit the records in this view using the **My Pipeline** filter. This effectively limits the records, so you only see your opportunities from your primary sales team. Removing the **My Pipeline** filter will allow you to see opportunities from other salespeople in your organization.

Creating a new opportunity

In Odoo 12, a potential sale is defined by creating a new opportunity. An opportunity allows you to begin collecting information about the scope and potential outcomes of a sale. These opportunities can be created from new leads, or an opportunity can originate from an existing customer.

For our real-world example, let's assume that Mike Smith has called and was so happy with his first order that he now wants to discuss using Silkworm for his local sports team's apparel. After a short conversation, we decide to create an opportunity by clicking on the **Create** button.



You can also use the + buttons within any of the pipeline stages to create an opportunity that is set to that stage in the pipeline.

In Odoo 12, the interface for entering an opportunity has been improved so you can enter the opportunity in a more concise pop-up panel:

The screenshot shows the Odoo 12 CRM Pipeline interface. At the top, there are tabs for CRM, Sales, Reporting, and Configuration, along with a search bar and administrator options. Below the tabs, the Pipeline section displays three stages: New, Qualified, and Won, each with a count of 0. A 'Create' button is visible above the stages. A 'My Pipeline' search bar is present. The main area shows a grid of opportunities. In the bottom-left corner, a modal window is open for creating a new opportunity named 'Sport Team Project'. The modal includes fields for Customer (set to 'Mike Smith'), Expected Revenue (\$2,500.00), and a rating of two stars. It also features 'Add', 'Edit', and 'Discard' buttons.

The preceding screenshot is of Odoo 12's new pop-up panel for entering an opportunity.

Opportunity

The title of your opportunity can be anything you wish. Naturally, it is important to choose a subject that makes it easy to identify the opportunity in a list. This is the only field required to create an opportunity in Odoo 12.

Customer

This field is automatically populated if you create an opportunity from the **Customer** form. You can, however, assign a different customer if you like. This is not a required field, so if you have an opportunity that you do not wish to associate with a customer, that is perfectly fine. For example, you may leave this field blank if you are attending a trade show and expect to have revenue, but do not yet have any specific customers to attribute to the opportunity.

Expected revenue

Here, you specify the amount of revenue you can expect from the opportunity if you are successful. Inside the full **Opportunity** form, there is a field in which you can specify the percentage likelihood that an opportunity will result in a sale. These values are useful in many statistical reports, although they are not required to create an opportunity.



Increasingly, more reports look at expected revenue and the percentage of opportunity completions. Therefore, depending on your reporting requirements, you may wish to encourage salespeople to set target goals for each opportunity to better track conversion.

Rating

Some opportunities are more important than others. You can choose zero, one, two, or three stars to designate the relative importance of a particular opportunity.

Looking at opportunities in your Pipeline

When you navigate to the **Sales** menu and choose **My Pipeline**, you will see your opportunities displayed in the Kanban view. Here, we see our brand new \$2,500 opportunity along with the customer, the next action we need to take, and when we need to take it. The following is a screenshot of the Kanban view of the pipeline:

The screenshot shows the Odoo CRM interface with the title bar "CRM" and tabs "Sales", "Reporting", and "Configuration". The main area is titled "Pipeline" with a search bar "My Pipeline" and a search icon. Below the search bar are buttons for "Create" and "Import", and a row of filter, group by, and favorite icons. The pipeline is divided into stages: "New", "Qualified", "Proposition", and "Won". Each stage has a progress bar and a count of 0. A specific opportunity card is selected, showing "Sport Team Project", "\$ 2,500.00, Mike Smith", and a rating of two stars. A small arrow icon is visible on the right side of the card.

Clicking on the small arrow on the Kanban card will bring up a small menu allowing you to perform actions related to the opportunity. You can edit the opportunity by clicking on it and choosing the **Edit** option:

The screenshot shows the Odoo CRM interface for creating a new opportunity. The main title is "Sport Team Project". Key fields include:

- Expected Revenue:** \$2,500.00
- Probability:** at 10 %
- Customer:** Mike Smith
- Salesperson:** Administrator
- Sales Team:** Sales
- Priority:** ★★☆
- Tags:** None

Below the form, there's a note section with a "Followup" tab selected. At the bottom, there are buttons for "Send message", "Log note", and "Schedule activity". A sidebar on the right shows recent activity: "Administrator - 4 minutes ago" created the opportunity, listing details like Customer, Revenue, Salesperson, Stage, Sales Team, and Active status.

This screen allows you to enter more information about the opportunity, including notes, email and phone contact information, when the opportunity is expected to close, and tags for helping to organize and locate your contacts by keyword.

Expected closing

When managing your opportunities, it is important to establish a goal for when you wish to close a sale. Providing an expected closing date is handy for managing opportunities and running reports, and for identifying which opportunities are due to be closed. The **Priority** setting ranges from lowest to highest, with three settings in between. In defining your CRM system, you should identify business rules for determining under what conditions an opportunity will receive the highest priority.

Tags

Odoo also allows you to assign multiple tags to an opportunity. For example, you could choose **trade show** and **sports** as tags to designate an opportunity that is sports-related and will take place at a trade show.

Email and Phone

The **Email** and **Phone** fields allow you to specify the primary contact methods you will likely use to communicate with your opportunity's contact person.

Internal Notes

The **Internal Notes** area is where you provide all the details on the opportunity. For our example, we kept the notes brief, but when you are working with real opportunities, make sure you take advantage of the **Internal Notes** area to document anything that will help you in closing the sale.

Contact information

When you create an opportunity from either a customer or a lead, the information is automatically brought over to the **Followup** tab in the **Opportunity** form:

Internal Notes	Contact Information
Customer Name	<input type="text"/>
Address	<input type="text"/> 444 South Main <input type="text"/> Street 2... <input type="text"/> Murphysboro, Illinois, 62966 <input type="text"/> United States
Website	<input type="text"/> e.g. www.odoo.com
Marketing	
Campaign	<input type="text"/>
Medium	<input type="text"/>
Source	<input type="text"/>
Misc	
Contact Name	<input type="text"/> Mike Smith <input type="text"/> Mister
Job Position	<input type="text"/>
Mobile	<input type="text"/>
Opt-Out	<input type="checkbox"/>
Referred By	<input type="text"/>

The preceding screenshot is of the **Followup** tab on the **Opportunity** form.

Address and contact information

The top half of the **Followup** tab contains the standard address and other contact information. This information is automatically populated if you choose a customer, but can be overwritten for this particular opportunity if you desire.



Odoo does not provide separate fields for first and last names in the same way as many other accounting systems. It is important to consider this as you plan how to organize customers in your system.

Mailings

The **Opt-Out** checkbox prevents the lead or customer associated with this opportunity from receiving mass mailings.

Marketing

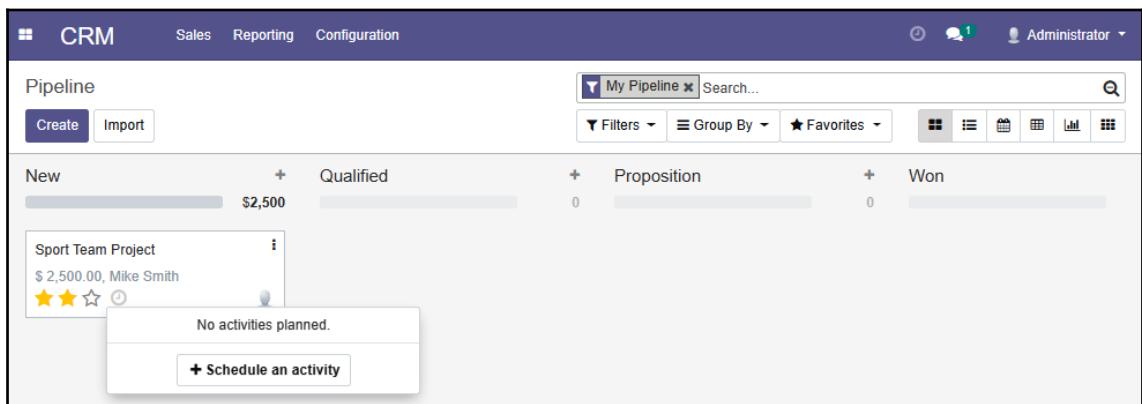
Also new in Odoo 12 is the ability to assign campaigns to an opportunity. Additionally, you can set the **Medium** (that is, phone, email, or television) and the source of the campaign.

Save the **Opportunity** by clicking on the **Save** button at the top of the form.

An introduction to sales stages

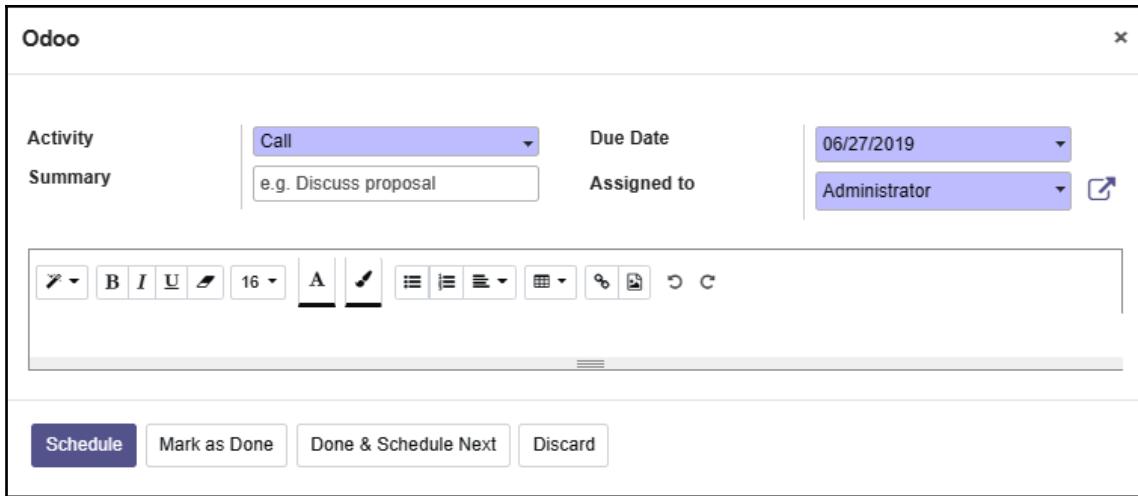
At the top of the Kanban view, you can see the default stages that are provided by an Odoo CRM installation. In this case, we see **New**, **Qualified**, **Proposition**, and **Won**.

As an opportunity moves between stages, the Kanban view will update to show you where each opportunity currently stands. Here, we can see this because **Sport Team Project** has just been entered into the **New** column:



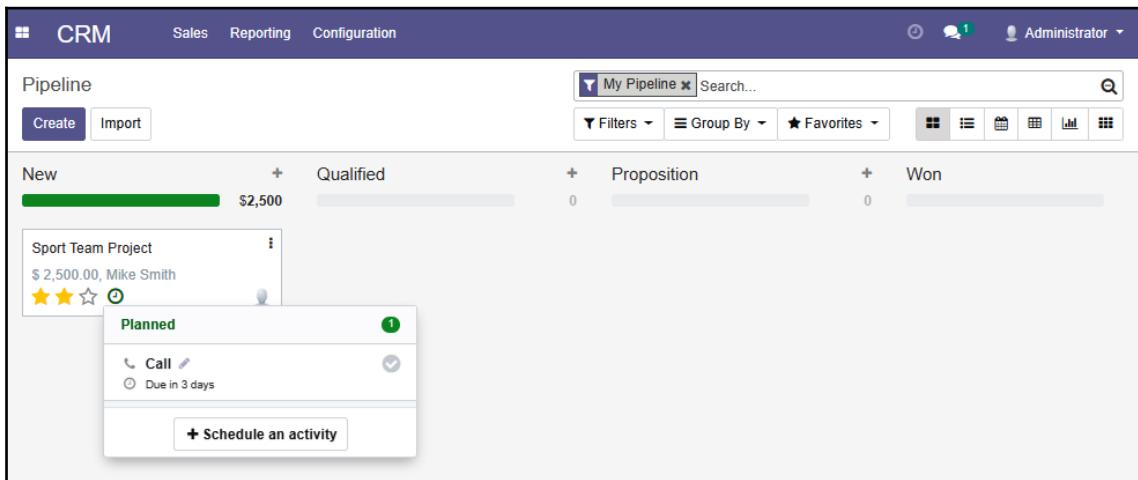
In Odoo 12, you can schedule activities from the Kanban view in the pipeline. This is one of the best methods for organizing communications related to your opportunities. To schedule an activity, click on the small clock icon on the Kanban card, and choose **Schedule an activity** from the drop-down menu.

Like most other actions, Odoo then brings up a form where you can provide the details of the activity:



Clicking on the **Schedule** button saves the activity and brings you back to the Kanban view. You can see the icon is now changed to green to indicate you have an activity that is scheduled.

If you click on the little green clock icon, you will see a small summary of the activity:

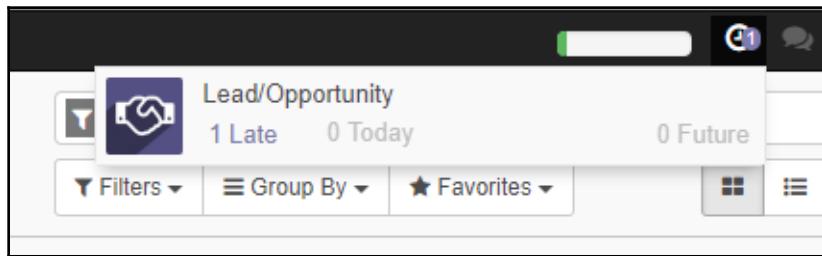


Here, we can see that an activity is planned and that it is a call that will take place within two days. This makes it easy to see what you need to do at a glance.

The Activities menu

In addition to being able to access the scheduled activity from the opportunity, you can also see a summary of activities within the **Activities** menu in the top-right menu bar. It resembles a little white or gray clock and may display a number if there are current activity notifications.

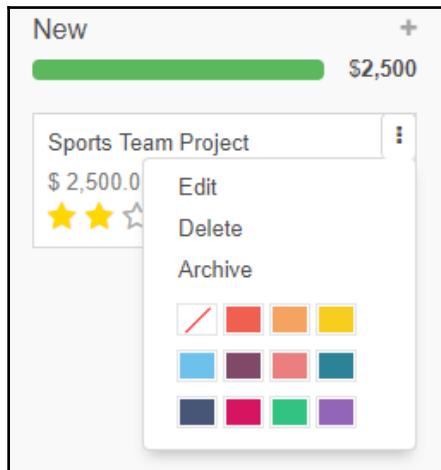
The **Activities** menu integrates with other applications as well. For example, if you have a purchasing procurement, you will also see it under your **Activities** menu:



Click on the clock icon to view the activity you just added.

Viewing the details of an opportunity

If you click on the three dots on the top-right side of the **Sports Team Project** opportunity card in the Kanban view, which appears when you hover your mouse over it, you will see a pop-up menu with your available options. The following screenshot shows the available actions in an opportunity:



Selecting the **Edit** option takes you to the opportunity record and into edit mode for you to change any of the information. In addition to this, you can delete the record or archive the record, so that it will no longer appear in your pipeline by default.

The color palette at the bottom lets you color-code your opportunities in the Kanban view.

The small stars on the opportunity card allow you to highlight opportunities for special consideration. Additionally, you can easily drag and drop an opportunity into other columns as you work through the various stages of the sale.

Using Odoo's Chatter feature

One of the biggest enhancements brought about in Odoo 7, and then expanded in later versions of Odoo, was the **OpenChatter** feature, which provided social networking-style communication for business documents and transactions. In Odoo 12, it is simply known as **Chatter**.

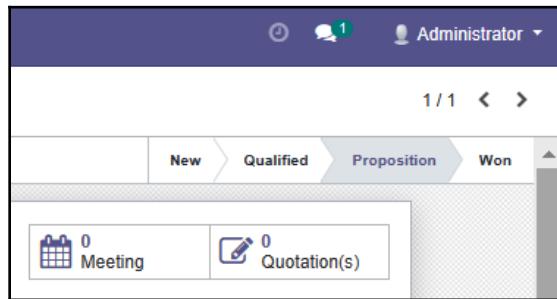
As we work on our brand new opportunity, we will utilize the **Chatter** feature to demonstrate how to communicate details between team members and generate log entries to document our progress.

The best thing about the **Chatter** feature is that it is available for nearly all business documents in Odoo. It also allows you to see a running set of logs of the transactions or operations that have affected the document. This means that everything that applies here to the CRM application can also be used to communicate about sales and purchasing, or to communicate about a specific customer or vendor.

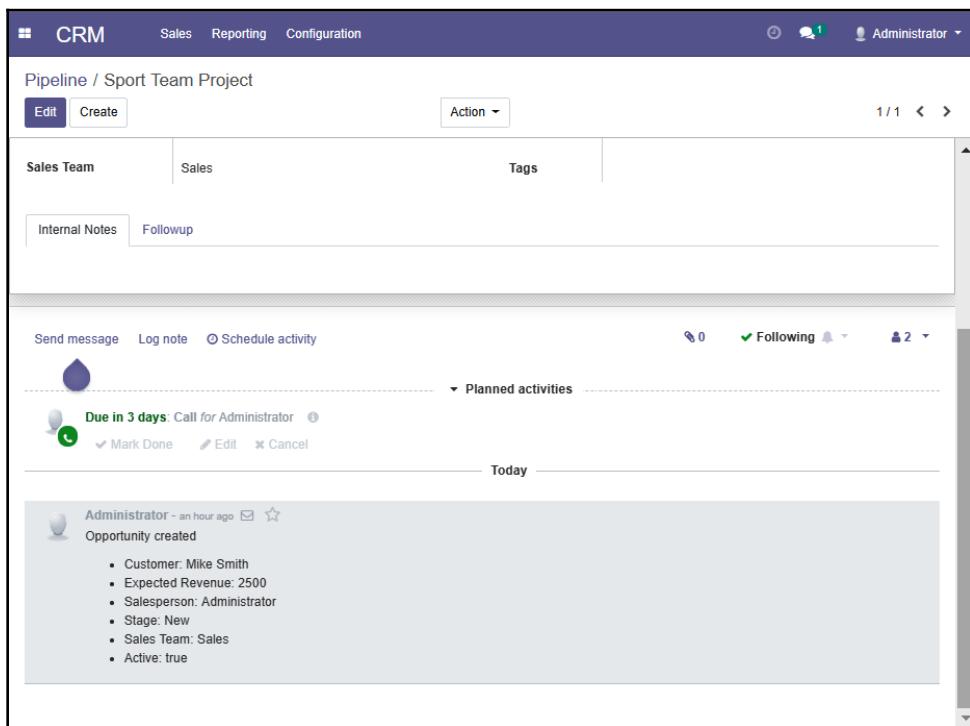
Changing the status of an opportunity

For our example, let's assume that we have prepared our proposal and made the presentation. Bring up the opportunity by using the right-click **Menu** in the Kanban view, or going into the list view and clicking on the **Opportunity** in the list.

It is time to update the status of our opportunity by clicking on the **Proposition** arrow at the top of the form:



Notice that you do not have to edit the record to change the status of the opportunity. At the bottom of the opportunity, you will now see a logged note generated by Odoo, which documents the changing of the opportunity from a new opportunity to a proposition. The following screenshot is of **Chatter** displaying a changed stage for the opportunity:



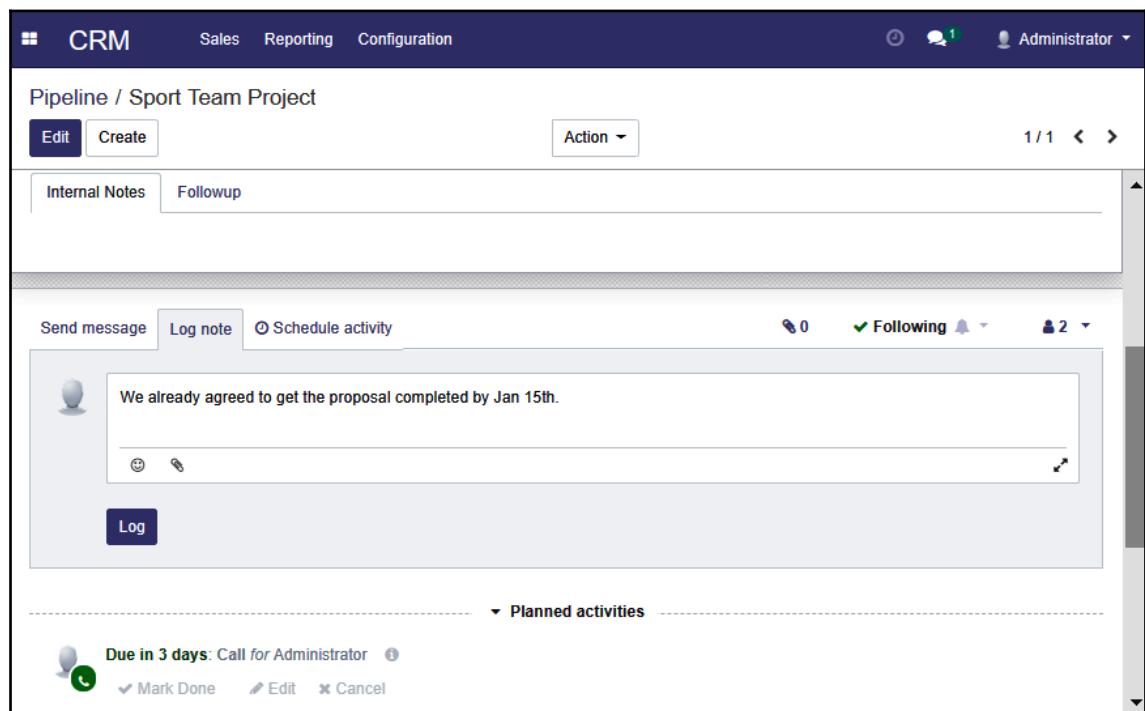
Notice how Odoo logs the events automatically as they take place.

Managing the opportunity

With the proposal presented, let's now take down some details from what we have learned that may help us later when we come back to this opportunity. One method of collecting this information could be to add the details to the **Internal Notes** field of the opportunity form. There is value, however, in using the **Chatter** feature in Odoo to document our new details.

More importantly, using **Chatter** to log notes gives you a running transcript with date and timestamps automatically generated. With the **Generic Notes** field, it can be very difficult to manage multiple entries. Another major advantage is that the **Chatter** feature can automatically send messages to team members' inboxes, updating them on the progress. So, let's see it in action!

Click on the **Log an Internal note** link to attach a note to our opportunity. The following screenshot is for creating a note:



The **Activity** option is unique to the CRM application, and will not appear in most documents. You can use the small icons at the bottom to add an emoji, attach a document, or open up a full-featured editor if you are creating a long note.

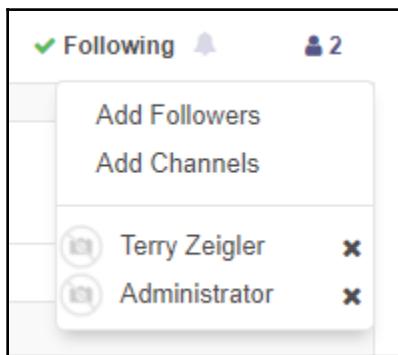


The full-featured editor also allows you to save templates of messages or notes that you may use frequently. Depending on your specific business requirements, this could be a great time saver.

When you create a note, it is attached to the business document, but no message will be sent to followers. You can even attach a document to the note by using the **Attach a File** feature. After clicking on the **Log** button, the note is saved and becomes part of the **OpenChatter** log for that document.

Following a business document

Odoo brings social networking concepts into your business communications. It is fundamental to this implementation that you can get automatic updates on a business document by following the document. Then, whenever there is a note, action, or a message created that is related to a document you follow, you will receive a message in your Odoo inbox. In the bottom-right-hand corner of the form, you are presented with the options for notifications and for adding or removing followers from the document. The following screenshot is of the **OpenChatter** follow options:

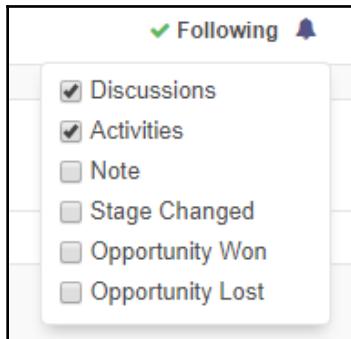


In this case, we can see that both **Terry Zeigler** and **Administrator** are set as followers for this opportunity. The **Following** checkbox at the top indicates that I am following this document. Using the **Add Followers** link, you can add additional users to follow the document.

The item's followers are notified by clicking on the little bell to the right of the **Following** button:



This brings up a list of the actions that will generate notifications for the followers:



The checkbox next to **Discussions** indicates that followers should be notified of any discussions related to this document. However, followers would not be notified, for example, if the stage changes.



When you send a message, by default, the customer will become a follower of the document. Then, whenever the status of the document changes, the customer will receive an email. You should test out all your processes before integrating with an email server. For additional resources on community modules that can help to manage Chatter features, please refer to the Appendix A, *Locating Additional Odoo Resources*, of this book.

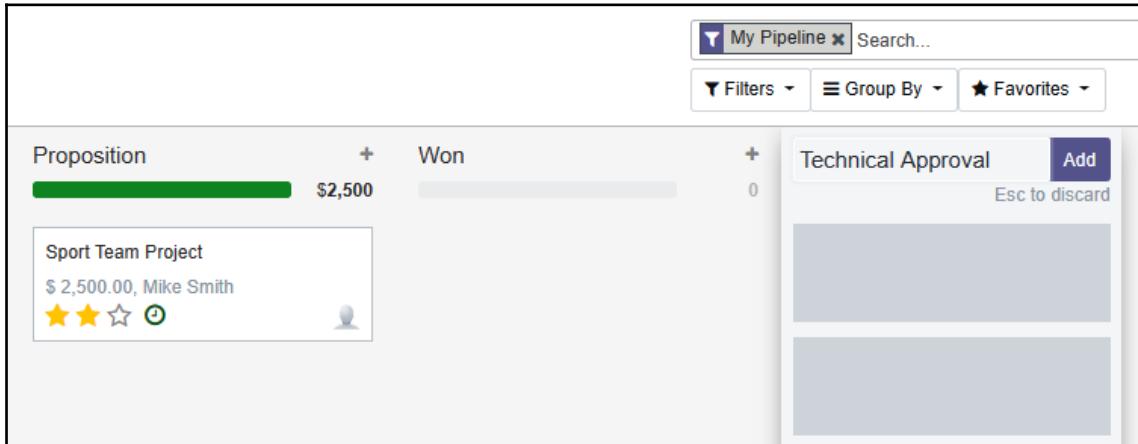
Modifying the stages of the sale

We have seen that Odoo provides a default set of sales stages. Many times, however, you will want to customize the stages in a way that can best deliver an outstanding customer experience. Moving an opportunity through stages should trigger actions that create a relationship with the customer and demonstrate your understanding of their needs. A customer in the qualification stage of a sale will have very different needs and expectations than a customer that is in the negotiation phase.

For our case study, there are sometimes printing jobs that are technically complex to accomplish. With different jerseys for a variety of teams, the final details need to go through an expert technical review and approval process before the order can be entered and verified.

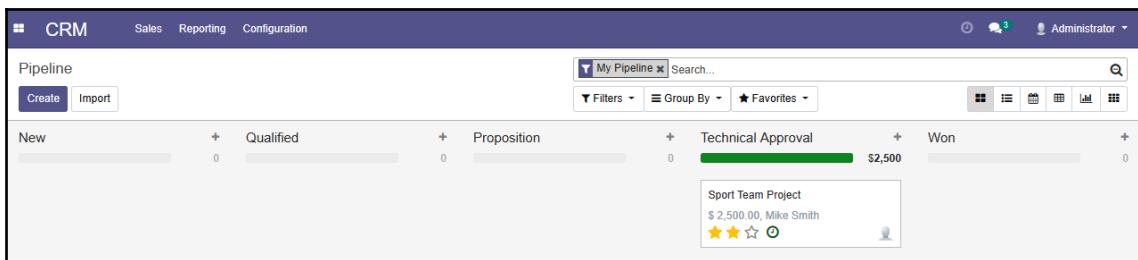
From a business perspective, the goal is not just to document the stage of the sales cycle; the primary goal is to use this information to drive customer interactions and improve the overall customer experience.

To add a stage to the sales process, bring up **Pipeline**, and then click on the **Add a Column** area on the right of the form to bring up a little pop-up form to enter the name for the new stage:



After you have added a column to the sales process, you can use your mouse to drag and drop the columns into the order that you wish them to appear. We are now ready to begin the technical approval stage for this opportunity.

Drag and drop the **Sport Team Project** opportunity over to the **Technical Approval** column in the Kanban view. The following screenshot is of the opportunities in Kanban view after adding the **Technical Approval** stage:



We now see the **Technical Approval** column in our Kanban view and have moved the opportunity over to that column. You will also notice that any time you change the stage of an opportunity, there will be an entry that will be created in the **Chatter** section at the bottom of the form. In addition to the ability to drag and drop an opportunity into a new stage, you can also change the stage of an opportunity by going into the **form** view.

Closing the sale

After a lot of hard work, we have finally won the opportunity, and it is time to turn it into a quotation. At this point, Odoo makes it easy to take that opportunity and turn it into an actual quotation.

So, open up the opportunity and click on the **New Quotation** tab at the top of the opportunity form:



After you have chosen **New Quotation**, you will be taken to a new quote form with the customer information already filled in:

A screenshot of the Odoo Sales Order creation form. The title bar says 'CRM', 'Sales', 'Reporting', 'Configuration', and 'Administrator'. Below the title is the sub-title 'Pipeline / Sport Team Project / New'. There are 'Save' and 'Discard' buttons. A toolbar below the sub-title has buttons for 'Send by Email', 'Print', 'Confirm', 'Preview', and 'Cancel'. A progress bar shows the process from 'Quotation' to 'Quotation Sent' to 'Sales Order'. The main area is titled 'New' and has a 'Customer' field set to 'Mike Smith'. It includes sections for 'Validity' and 'Payment Terms'. Below these are tabs for 'Order Lines', 'Optional Products', and 'Other Information'. The 'Order Lines' tab is active, showing a table with columns: Product, Description, Ordered Qty, Unit Price, Taxes, and Subtotal. Buttons at the bottom of this section allow adding a product, a section, or a note. At the bottom right, there is a summary: 'Untaxed Amount: \$ 0.00', 'Taxes: \$ 0.00', and 'Total: \$ 0.00'. A note at the bottom left says 'Terms and conditions... (note: you can setup default ones in the Configuration menu)'.

In the preceding screenshot, we can see the customer details are already filled.

Your opportunity converted to a quotation

The workflow in Odoo handles moving over all the required information from your opportunity to your quotation document. At this point, you are ready to begin adding line items and creating a quotation just like we did in [Chapter 2, Installing Your First Odoo Application](#).

Please note that just because you create a new quotation from an opportunity, Odoo still leaves the opportunity open. Therefore, you must go back into **Pipeline** and mark the opportunity as **Won** if you are truly done with the opportunity.

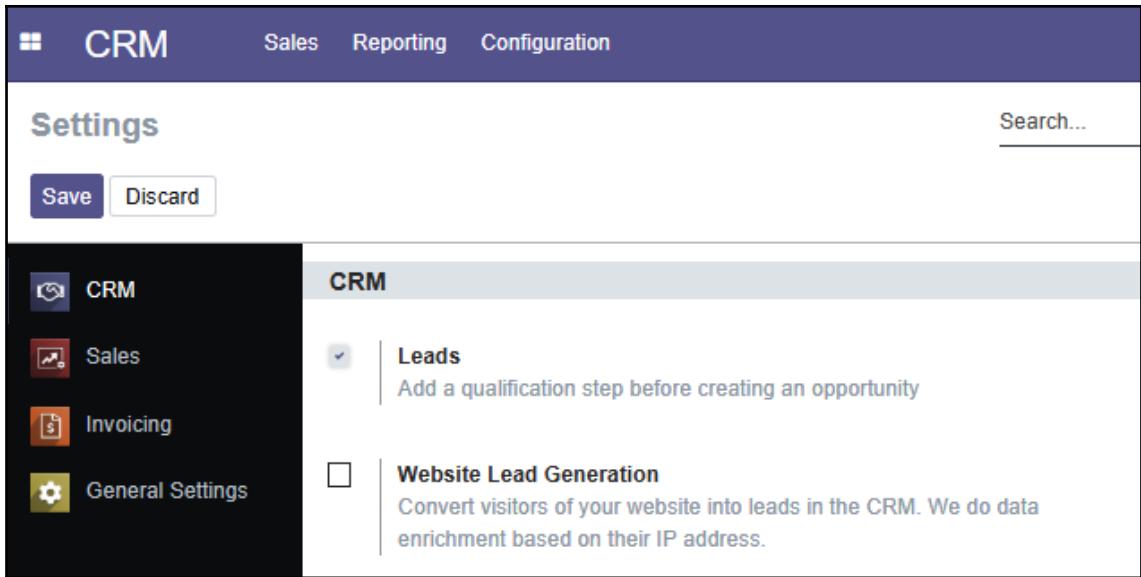
Leads and opportunities

Odoo provides two primary documents for managing interactions with your customers or potential customers. You can think a lead as less critical, and perhaps less likely, to turn into a real sales situation than an opportunity. A good example of leads would be that you get a few dozen business cards from people you meet at a conference. You could add each of them as a lead for further follow up. An example of an opportunity would be if you met someone at a conference and had a detailed conversation about how your company provides appropriate services.

Many people get confused between when to use leads and when to use opportunities. The best way to remember is that leads are intangible and are, essentially, potential contacts. Opportunities should be more clearly defined, have some sort of expected income if they are successful, and provide significant project details and scope compared to a simple lead.

Turning on leads in Odoo 12

When you first install the Odoo CRM application, leads are turned off by default. You can enable leads by first choosing **Settings** under the **Configuration** section in the **CRM** menu. In the **Settings** form under **Leads**, you can turn on **Leads** by selecting the second option and then clicking on **Save**:



Once you have applied the changes in **Settings**, Odoo will refresh and add a **Leads** option to your **CRM** menu.

Creating leads in Odoo

Many times, it can take quite a bit of work to uncover an opportunity. In Odoo, you create leads when you need a qualification step before creating an opportunity or a customer. For example, you may receive a business card or an unqualified lead from your website. Another common situation is that leads are perhaps purchased from a mailing list and then imported into Odoo.

Let's create a new lead for a potential customer we met at a local event.

Under the **CRM** menu, click on **Leads** and then click on the **Create** button to open a new lead. The following screenshot is of the form used for creating a new lead:

The screenshot shows the Odoo CRM interface for creating a new lead. The title bar indicates it's a new Odoo session at localhost:8069. The main header shows 'CRM' and the sub-header 'Leads / New'. Below the header, there are two buttons: 'Save' (highlighted in blue) and 'Discard'. The main content area is titled 'Lead' and contains the following fields:

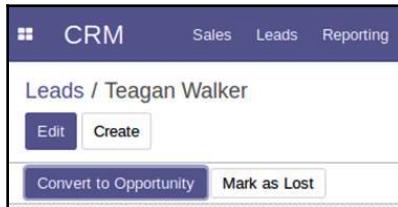
- Company Name:** Creative Hair Designs
- Address:** 440 South Hampton
Street 2...
Carbondale, Illinois 62901
United States
- Contact Name:** Teagan Walker (Miss)
- Email:** teaganwalker@example.com
- Job Position:** (empty)
- Phone:** 444-555-6666
- Mobile:** (empty)
- Website:** e.g. www.odoo.com
- Salesperson:** Terry Zeigler
- Sales Team:** Sales
- Priority:** ★★★
- Tags:** Design

At the bottom of the form, there are tabs for 'Internal Notes' and 'Extra Info'. The 'Internal Notes' tab contains the note: 'Linda has expressed interest in using our services to promote our business'. At the very bottom of the page, there are buttons for 'Send message', 'Log note', 'Schedule activity', 'Follow' (with a count of 0), and a small icon.

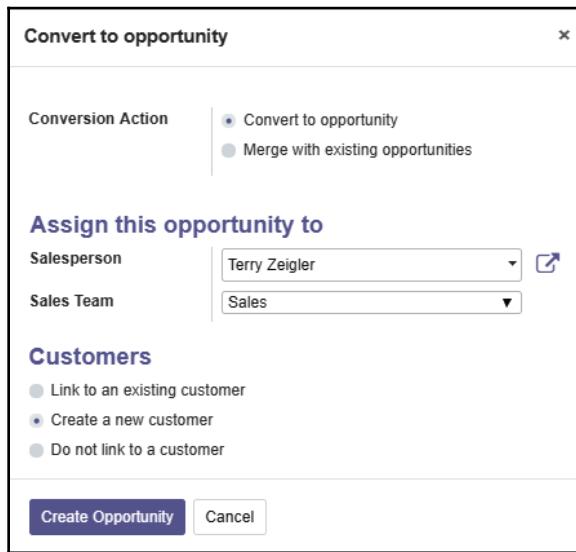
As you can see in the preceding screenshot, the form is very similar to the standard customer screen. There is a good reason for this, as Odoo uses a standard structure to hold address information for leads, customers, suppliers, and users or employees. In our example, we have filled out the basic contact and address information and assigned our sales representative to this lead.

Converting a lead into an opportunity

Leads will stay leads indefinitely until you take some action to either turn them into opportunities or mark them as **Lost/Dead**. You will notice that on the top-left of your form there is a button labeled **Convert to Opportunity**. At any point, you can convert a lead into an opportunity by simply clicking on this button:



Once you click on **Convert to Opportunity**, you will be presented with an Odoo wizard that will allow you to choose how you wish to handle converting the lead into an opportunity:



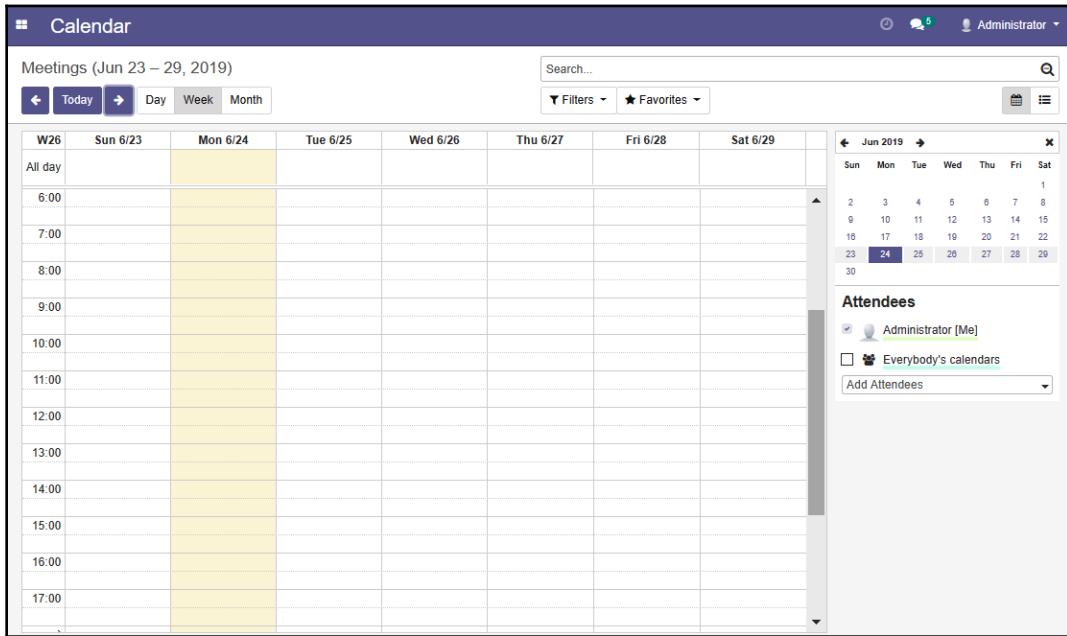
Each of the options presented is relatively self-explanatory. The conversion action determines whether you will create a new opportunity or merge this lead with an existing opportunity. You also get the option of assigning the opportunity to a specific salesperson.

Finally, you can tell Odoo whether you wish to create a new customer for this opportunity, or whether, instead, you wish to assign this opportunity to an existing customer.

Using Odoo to schedule calls, meetings, and events

Often, when working with leads and opportunities, you will find it beneficial to schedule meetings and calls. Odoo provides a built-in meeting scheduler that you can use specifically to manage your schedule and relate those events to customers within Odoo. Odoo considers it so useful that they have a dedicated menu for it! Let's take a look at how we can schedule an event in Odoo. Meeting scheduling is handled in the **Messaging** menu of Odoo. Begin by choosing **Calendar** from the main menu.

Odoo will then display your personal calendar, as follows:

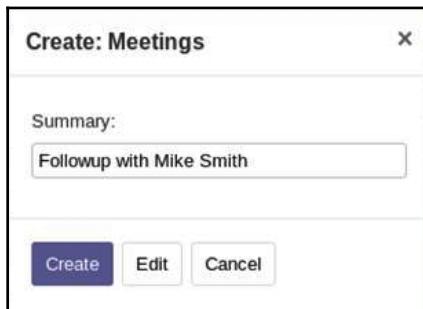


Odoo will bring up the current week. Arrows on the top-left side of the form allow you to quickly navigate to the previous and next months, respectively. On the right-hand side of the title, there is an option to look at the calendar by week or by day. This can be particularly valuable for seeing more information when you have many meetings scheduled.

On the far-right-hand side, you have a small calendar for the next month. This small calendar is interactive, and you can use it to quickly jump to that month or even a specific day.

Scheduling an event

Scheduling an event is very easy. Simply click on the day that you wish to schedule an event. You will then be prompted to give a name for the event, and to either create or edit the event:



After entering the event **Summary**, you can quickly **Create** the event, or you can choose **Edit** to provide additional details about the event.

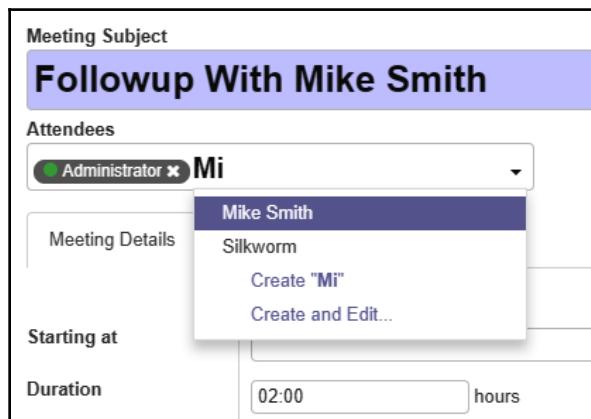
Depending on how you want to organize and manage your meetings, it may work for you just fine to create the event, provided the event **Summary** is enough information for you to take the action you require. Typically, however, it will be better practice to **Edit** this event and provide some additional details.

Click on **Edit** to create an event and automatically bring it up for editing:

Odoo will automatically bring over the event **Summary** that you filled in after clicking on the day. Notice, however, that instead of event **Summary** the title is now **Meeting Subject**. Perhaps Odoo will modify this in the future for greater consistency.

Adding attendees to your meeting

If you have created a meeting without first opening a customer record, you may not have them added as an attendee of the meeting. Also, you may wish to add additional people to the meeting. When you are meeting with a client, customer, or vendor, it is largely up to you if you wish to add the attendee here in the list. For example, you could add Mike Smith to the list of attendees if he were not already included. Odoo will automatically search in real time as you type out the name:



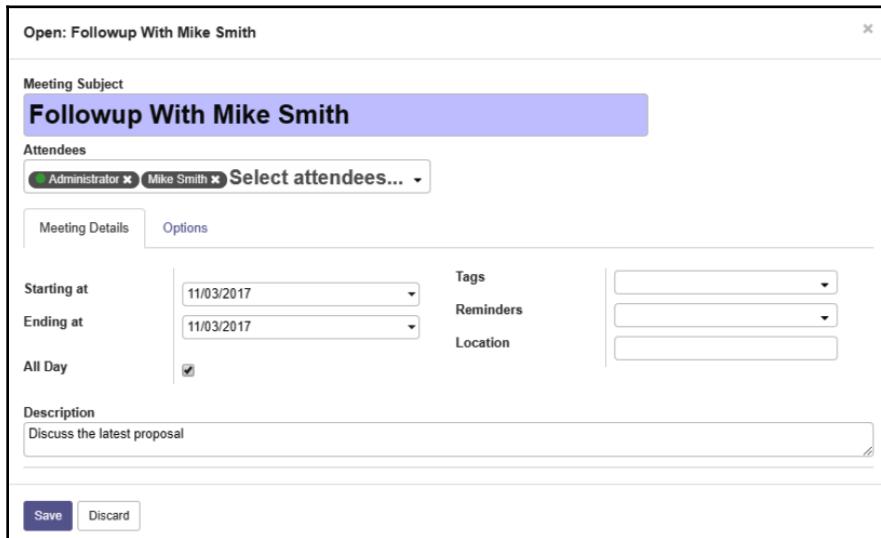
Odoo will then add the attendee to the **Attendees** list. For internal communications, this can be used to make sure that all the necessary team members are notified of the meeting if they are also using the Odoo schedule.



Odoo will provide you with a warning if you add an attendee that does not have an email address. In this case, Odoo will still add the attendee but, naturally, any automated notifications cannot be delivered to the attendee via email.

Specifying meeting details

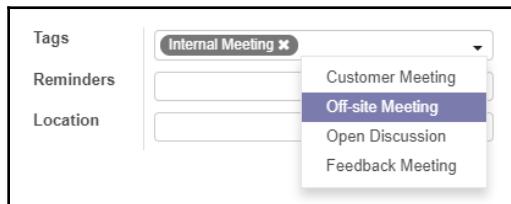
Odoo's meeting scheduler offers quite a few different options that assist you in customizing a meeting. One of the first things to notice is that, by default, Odoo schedules a meeting for the full day. If your meeting does not have a specific time, you can check the **All Day** option. After you have checked the option, the **Duration** field disappears and is replaced with an **Ending at** field to specify the ending day for the event:



This allows you to specify events that span several days.

Specifying tags for your meeting

Odoo provides a set of default meeting **Tags** that quickly tell you the overall scope of the meeting:

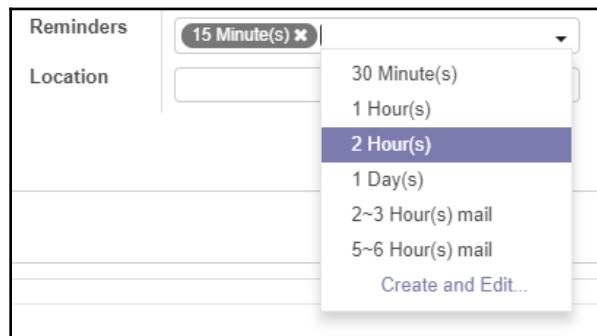


You can specify multiple tags and create new tags to organize your meeting schedules.

Setting up reminders for your meeting

Often, you may want to have a notification or a reminder appear a short while before your meeting. Setting **Reminders** can help to prevent you missing an important meeting. Odoo offers two kinds of basic reminders: notifications and email reminders.

Notifications will prompt you visually onscreen in the top-right corner of your window when the time before the meeting is reached. Email **Reminders** will send you an email message at a specified time:

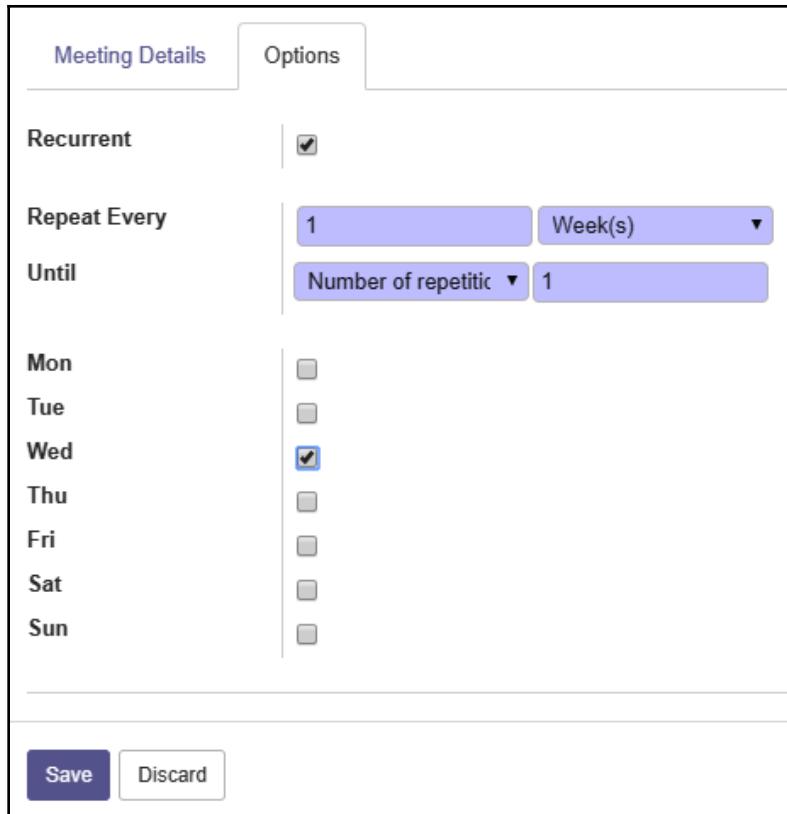


In addition to **Tags** and **Reminders**, you also have the option to specify the location of the meeting. This location is a simple text field and is just some extra information that you can use to keep your team members informed.

Specifying additional meeting options

On the **Options** page, Odoo allows you to specify several additional options for meetings. One of the most powerful features is the ability to configure recurring meetings. When you select the **Recurrent** option, additional options will become available in which you can select the interval.

Depending on the interval you select, the form will refresh with the appropriate options for that interval. In the following screenshot, we have selected a weekly interval. Odoo then allows us to select on which day(s) of the week the meeting will repeat.



In addition to selecting the specific days, you can also specify how long you have until the recurring meeting will end. The preceding example demonstrates how you can specify a meeting to end based on the number of repetitions. If you choose, you could instead select an end date to stop the recurring meeting.

Summary

In this chapter, we started by discussing the role of a CRM system in a modern-day business. We installed the CRM module, created salespeople, and proceeded to develop a system to manage the sales process. In our example, we walked an opportunity through the various stages in the sales process. Finally, we learned how to modify stages in the sales cycle and turn an opportunity directly into a quotation.

In the next chapter, we will turn our attention to purchasing products and setting up the Manufacturing application to handle production operations.

4

Purchasing with Odoo

In this chapter, we start getting into what could be considered the core functionality of most **Enterprise Resource Planning (ERP)** systems. We will begin by setting up a vendor and then purchasing raw-material components. After the products arrive, we will receive the products in our inventory and pay the invoice in order to complete the purchasing cycle.

The topics we will cover in this chapter include the following:

- Examining a typical purchasing process for a business
- Setting up your vendors and warehouse locations
- Entering a quote and turning it into a purchase order
- Receiving products from your vendors
- Paying invoices
- Understanding units of measure for placing bulk orders

Understanding the overall purchasing process

Let's begin by taking a bird's eye view of the purchasing process. Putting together a purchasing system requires several steps, and initially, it can be confusing for people who are new to ERP systems. But when you break the steps down and look at them individually, the process becomes much easier to understand.

Setting up a vendor

When you set up a vendor, you are determining the individuals or companies that are providing you with products. Sometimes, vendors are also referred to as suppliers. In Odoo, it is completely possible to create a product and sell it without implementing a purchasing system. However, to begin using your system for purchasing, you will need to set up the vendors.

The steps you take to set up a vendor are much the same as those for setting up a customer. In fact, now is as good a time as any to reiterate that Odoo maintains core customer, employee, and vendor records all in the same model (or table), named `res.partner`. Odoo distinguishes between customers, vendors, and those who are both, with the use of the **Is a Customer** and **Is a Vendor** checkboxes.

Setting up warehouse locations

Once you have decided to start using Odoo to purchase your products, you will need to set up locations to receive them from. In a simple small business, you may only have one location, but other companies may have hundreds of warehouse locations. In Odoo, each location can maintain its own address, and it is possible to create nested sub-locations for better management and reporting of inventory.

Generating quotations and purchase orders

To acquire the raw product, you will need to create **Requests for Quotations** and/or **Purchase Orders** to send to your vendors. In purchasing, these are the documents you create that tell the vendors which products you require, the quantity in which you require them, and what you expect to pay for those products. Often, this process is referred to as **procurement**. Depending on the industry and the specific location of the company, it is possible that there might be a variety of methods for managing quotations and approvals when purchasing products.

Receiving the product

In a simple purchasing workflow, once your purchase order has been received by the vendor, you will be waiting for them to fulfill the order. At some point, you will receive the product. Depending on your industry, this could be on the same day, or could even stretch to months. When the delivery is complete, you will receive the products and they will be delivered to the location you select. Now that the product has been received, you are ready to create a manufacturing order. But first, let's pay the vendor for what you ordered.

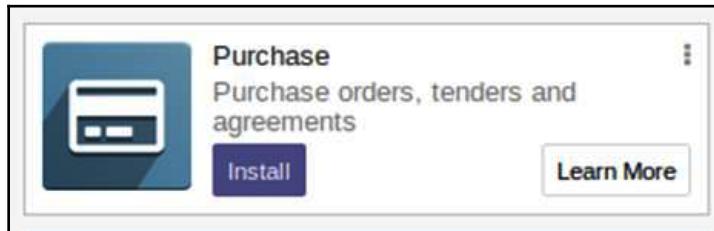
Settling the invoice

Once you have received the product, it is just a matter of time before you must pay for it. Invoicing can happen at the time you order the product; before the product is shipped to you; or after you have received the product. Regardless of when you get an invoice, you can be sure that if you are receiving products, you will eventually be invoiced for them.

When an invoice is received, it is essential to compare it to the purchase order for accuracy. Any discrepancies between the purchase order and invoice must be resolved before the invoice is paid. Essentially, this is your way of ensuring that you are only paying for the products you have authorized for purchase. Finally, it is good practice to match the receiving or delivery order to the purchase order *and* to the invoice as well. This *three-way match* ensures that you got exactly what you ordered and that the invoice reflects exactly what you are required to pay.

Installing the Purchase application

Odoo is a modular set of applications for which you only install the applications you need. Therefore, we must install the **Purchase** application in order to continue. By this point, you should be familiar with the process of installing a new application in Odoo. The following is a screenshot of the **Purchase** application as it appears in the **Apps** list:



When you install the **Purchase** application, you will get a new purchasing menu in your toolbar.



It is possible that these menus may already exist, or that purchasing may already be installed if you have installed another module, such as **eCommerce**, that requires **Purchase Management** as a dependency.

The **Purchase** menu is where you can create quotations and purchase orders for the products you purchase from your vendors.

Setting up your first vendor

To begin setting up your first vendor, you should select **Purchase | Vendors** and click on **Create**:

The screenshot shows the Odoo web interface for managing vendors. At the top, there's a dark blue header with the 'Purchase' module name. Below the header, the main content area has a title 'Vendors' and two buttons: 'Create' and 'Import'. To the right of the buttons are several small icons for filtering, grouping, and favoriting. The main body of the page is mostly empty, with a large placeholder image of a smiling notepad character and the text 'Create a new vendor in your address book'. Below this, a smaller note says 'Odoo helps you easily track all activities related to a vendor.'

This is the **Vendors** listing, but as it is empty, you will see instructions on how you can add a new vendor. Odoo also lists a few of the features that you can expect from vendor management, such as tracking discussions, history of purchases, and documents associated with a vendor.

After clicking **Create**, Odoo will bring up the **Vendors** form for you to fill out:

The screenshot shows the Odoo interface for creating a new vendor record. The top navigation bar includes 'Purchase' (selected), 'Purchase', 'Control', 'Reporting', and 'Configuration'. On the right, there are icons for a timer, a speech bubble with a '2', and 'Administrator'. Below the navigation is a sub-menu bar with 'Vendors / New' and buttons for 'Save' and 'Discard'.

The main form area has a title 'T-Shirt Supply Co.' with a camera icon. It includes two radio buttons: 'Individual' (unchecked) and 'Company' (checked).

The form fields are organized into sections:

- Address:** 564 Pine Street, Street 2..., Canton, Ohio (US) 44702, United States.
- Tax ID:** e.g. BE0477472701
- Phone:** 444-555-6666
- Mobile:** (empty)
- Email:** (empty)
- Website:** e.g. www.odoo.com
- Language:** English
- Tags:** (empty)

At the bottom of the form are tabs: 'Contacts & Addresses' (selected), 'Internal Notes', 'Sales & Purchases', and 'Invoicing'. There is also a 'Add' button.

This form is very much like the customer form in that it is based on the same basic structure. In fact, it is perfectly acceptable for a customer to also be a vendor. When you create a new vendor record, a vendor checkbox is automatically marked for you under the **Sales & Purchases** page on the form. Sometimes, this can get a little confusing for people who are new to Odoo; therefore, this chapter will start to make the relationships between companies, contacts, customers, and vendors in Odoo clearer.

Designating a vendor as Individual or Company

Much like when you set up a customer, the **Individual** and **Company** options at the very top of the form are where you inform Odoo of the relationship you have with this vendor. Typically, you will be purchasing products from a company.

By default, Odoo will automatically set the designation to **Company**, as it is more likely that companies are going to be making their purchases from other companies, instead of from individuals. For our example, we will leave the **Company** option checked.

Once you've filled in the vendor name, address, and other contact information, as well as the required accounting information, click **Save** at the top of the form.

Product type

Next, we want to link a product to our vendor. Select **Purchase | Products** and then click on the product we created in *Chapter 1, Setting Up Odoo 12—Medium White T-shirt*.



When you configure purchasing, you will want to pay special attention to product type. In previous editions of Odoo, when you installed the Purchasing application, the Inventory application was also installed along with it. This gave access to additional options when purchasing, such as routing and the ability to designate a product as stockable. In Odoo 12, you can designate only **consumable** or **service** as the available stock types.

Setting the cost price of the product

Often, you will wish to assign a cost to the product. This will be the cost that will appear in your purchase quotations, though it can be overwritten at any time to reflect a vendor's new pricing. If your vendor happens to give you a one-time discount, you will want to reflect that change on the actual purchase order, rather than here, in the base product record. For our example, we have set the **Cost** price of the shirt to 7.25.

Assigning vendors to the product

You can use the **Purchase** tab to bring up the list of vendors for the product. It is very common that a company may have multiple vendors that offer the same product. Click on **Edit** then choose **Add a Line** in the vendor grid in order to add the vendor to the product:

The screenshot shows the Odoo Product Management interface for a 'Medium White T-Shirt'. The product details include a camera icon, a blue box labeled 'Medium White T-Shirt', and two checked checkboxes: 'Can be Sold' and 'Can be Purchased'. A summary box at the top right shows stock levels: 24.00 Unit(s) Purchased, 11.00 Unit(s) On Hand, 10.00 Unit(s) Forecasted; 0 Product Moves, 0 Reordering R..., 0.00 Unit(s) Sold; and 0 Bill of Materials. Below this are tabs for General Information, Variants, Sales, Purchase (selected), and Inventory.

Vendors

Vendor	Product Variant	Minimal Quantity	Unit of Measure	Price	Start Date	End Date
T-shirt Supply Co.		12.00	Unit(s)	7.25		

Add a line

Vendor Bills

Vendor Taxes: Tax 15.00% (dropdown menu)

Control Policy: On ordered quantities (radio button)

Description for Vendors

This note will show up on purchase orders.

Establishing the vendor

You have the choice in the drop-down list to search for vendors, as well as to create and edit a new vendor on the fly. To the far right of the drop-down list, you can use the small icon to edit the current vendor:

The screenshot shows the 'Create Vendors' dialog box. It has two main sections: 'Vendor' and 'Price List'.

Vendor

- Vendor: T-Shirt Supply Co. (dropdown menu)
- Vendor Product Name: MED WHT SHRT
- Vendor Product Code: MWT-20
- Delivery Lead Time: 4 days

Price List

- Minimal Quantity: 12
- Price: 7.25
- Validity: From 01/03/2019 to 01/31/2020

At the bottom are four buttons: Save & Close, Save & New, Discard, and Remove.

In the drop-down list, we have selected **T-Shirt Supply Co.** as the company to provide our **Medium White T-shirt** items.

Designating the vendor product name and product code

Because a vendor may describe a given product using different product codes or product names than your company does, here, you have the option to specify how the vendor identifies the product. This information will be displayed on the purchase quotations and purchase orders you create in order to make sure you get the right product from the vendor.

Setting minimal quantity

Vendors will often have a minimum order quantity for a product. Sometimes, vendors may actually offer to sell you a lower quantity, but the cost per unit is dramatically higher. Setting a minimum quantity in this form allows you to prevent those problems by forcing purchase quantities to be at least the minimum quantity value. For our example, we will set the minimum quantity to 12.

Calculating delivery time

Depending on the vendor, a product may take less or more time to obtain. Often, this can make a difference when you are putting together a time-sensitive purchase quotation. A product may be cheaper, but if the delay is too long and puts the delivery time in jeopardy, you may need to buy the product at a higher price from another vendor who can deliver the product faster. Setting the delivery time in days for the vendor to deliver the product provides your purchasing agents with the information they require to make decisions based on price and availability. For our example, we have set the delivery lead time to 4 days.

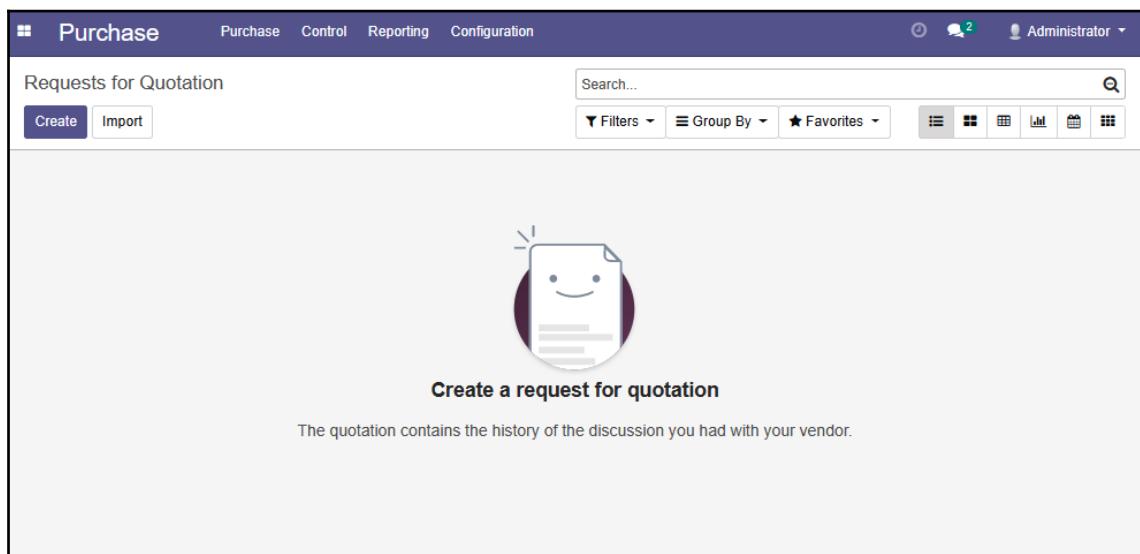
Setting price and validity

Now, in Odoo 12, you can also specify the price for the item from that specific vendor. This simplifies managing your vendor pricing. When you select the given item for a specific vendor, you will get the price you have specified here. Optionally, you can specify validity dates, so that you can proactively manage products when suppliers bring on new products or perhaps discontinue a product.

Creating your first purchase quotation

Now that we have our vendor entered and the product associated with the vendor, we are ready to create our first **Request for Quotation (RFQ)**. This is typically the document you will create when requesting pricing from a vendor (sometimes called a supplier) prior to actually ordering the product. For our example, we are going to create a request for a quotation for one dozen **Medium White T-Shirts**:

1. Go to **Purchase | Requests for Quotation** and click on **Create** to make a new RFQ, as shown in the following screenshot:



2. After you click on **Create**, the RFQ will appear for you to enter the required information:

New

Vendor	T-shirt Supply Co.	Order Date	05/30/2019 02:35:39			
Vendor Reference						
<input type="button" value="Products"/> <input type="button" value="Other Information"/>						
Product	Description	Scheduled Date	Quantity	Unit Price	Taxes	Subtotal
+ Medium White T-Shirt	Medium White T-Shirt	06/03/2019 02:35:39	12.000	7.25	Tax 15.00%	\$ 87.00
<input type="button" value="Add a line"/>						
<input type="text" value="Define your terms and conditions ..."/>				<input type="text" value="Untaxed Amount: \$ 87.00"/> <input type="text" value="Taxes: \$ 13.05"/> <input type="text" value="Total: \$ 100.05"/>		

When you first create the request for quotation, the **Order Date** will be automatically populated with the current date. You can then select the vendor you wish for the RFQ, as well as include an optional vendor reference and source document. An example of a source document would be a sales order number that triggered the purchase order process.

Adding products to your request for quotation

After you click **Add a Line**, you can select the product from the drop-down list on the far left. The description will automatically be filled in. The scheduled date will be determined based on the delivery delay from the vendor. We also find that the minimum order quantity from the vendor has been pulled to the RFQ. Finally, the unit cost is populated from the vendor unit cost we had specified.

Printing RFQs and updating status

For now, we will skip the **Deliveries & Invoices** and go right to printing our RFQ. By default, Odoo will print to a PDF file. This file can easily be attached to emails.

Once you have configured an email server, you can configure Odoo to automatically send the purchase order by email.

Confirming a purchase order

Once you have a final quotation, you are ready to confirm the purchase order. It is very important to understand that once you have confirmed the RFQ, it becomes a purchase order and it can no longer be modified. Once you are sure you wish to finalize the purchase order, click the **Confirm Order** button. Any modifications that need to be made at this stage would require you to duplicate and cancel the original order. This is necessary so that Odoo can maintain an audit trail.



If you happen to receive an error message reading **No Expense Account** when you attempt to confirm the order, check your settings for your chart of accounts. You must have an expense account designated for the products contained on the purchase order.

The following screenshot shows an example of a confirmed **Purchase Order**:



Once you have confirmed the **Purchase Order**, the form will refresh to show the new status of the purchase order. At this point, you are waiting on the vendor to deliver the products and send you an invoice.

Receiving products

When you installed purchasing in previous versions of Odoo, it was possible to designate products as stockable, receive them in your inventory, and manage many details. Now, in Odoo 12, the receiving of products is much simpler if you do not have inventory installed.

To receive products, you simply go to the **Purchase Order** and click **Edit**. Then, go in and specify the quantity you have received:

Product	Description	Scheduled Date	Quantity	Received Qty	Billed Qty	Unit Price	Taxes	Subtotal
Medium White T-Shirt	Medium White T-Shirt	06/03/2019 02:35:39	12.000	12.000	0.000	7.25	Tax 15.00% <input checked="" type="checkbox"/>	\$ 87.00 <input checked="" type="checkbox"/>

Define your terms and conditions ...

Untaxed Amount: \$ 87.00
Taxes: \$ 13.05
Total: \$ 100.05

And just like that, you have marked the goods in purchasing as you have received.

Creating a vendor bill

Once you have received your product, sooner or later, you should receive a bill. You can create one for this purchase order directly by clicking **Create Bill** at the top of the form. In fact, Odoo will have colored the button purple to signify that it is the next step you should most likely take in the workflow.

When you click **Create Bill**, Odoo will automatically create a new vendor bill and add the purchase order as a line item detailing the expense:

Draft Bill - First Number:
BILL/2019/0001

Vendor	T-shirt Supply Co.	<input checked="" type="checkbox"/> Source Document	PO00001		
Vendor Reference		Bill Date			
Auto-Complete	Select a purchase order or an old bill	Due Date	06/03/2019		
		Bank Account			
<input type="radio"/> Bill		<input type="radio"/> Other Info			
Product	Description	Quantity	Unit Price	Taxes	Amount
+ Medium White T-Shirt	PO00004: Medium White T-Shirt	12.000	7.25	Tax 15.00%	\$ 87.00
Add a line					
Tax Description		Tax Amount		Untaxed Amount:	
Tax 15.00%		\$ 13.05		\$ 87.00	
Add a line				Tax:	
				\$ 13.05	
				Total: \$ 100.05	

Alternatively, you can go in and simply create a vendor bill from an invoice you have received, and then look up the purchase order to apply it to.

Paying vendor bills without the purchase order

This is often called a supplier or vendor invoice. It is, of course, possible to receive a bill before you receive products. Each business will have to decide the exact workflow for when they pay bills and under what conditions. Here are the steps that you must take in order to create a vendor bill:

1. To create a vendor bill, go to **Purchase | Control | Vendor Bills**:

The screenshot shows the Odoo Purchase Control Vendor Bills interface. At the top, there's a navigation bar with tabs for Purchase, Control, Reporting, and Configuration. On the left, there's a sidebar with a 'Vendor Bills' section containing 'Create', 'Import', and 'Upload' buttons. The main content area has a search bar and filter options. In the center, there's a large button with a smiley face icon and the text 'Create a vendor bill'. Below this, there's a descriptive text: 'Use this menu to control the invoices to be received from your vendors. When registering a new bill, set the purchase order and Odoo will fill the bill automatically according to ordered or received quantities.' The entire interface is presented within a light gray box.

2. Click **Create** to bring up the **Vendor Bills** form:

The screenshot shows the 'Draft Bill' section of the Odoo Vendor Bills form. It includes fields for Vendor (dropdown), Vendor Reference (text input), Auto-Complete (dropdown with placeholder 'Select a purchase order or an old bill'), Bill Date (dropdown), Due Date (dropdown), Currency (dropdown with 'USD' selected), and Bank Account (dropdown). Below these are tabs for 'Bill' (selected) and 'Other Info'. A table for adding bill lines has columns: Product, Description, Account, Quantity, Unit of Measure, Unit Price, Taxes, and Amount. A note 'Add a line' is present. At the bottom, there's a table for taxes with columns: Tax Description, Tax Account, and Tax Amount. A note 'Add a line' is present. Summary totals are shown: Untaxed Amount: \$ 0.00, Tax: \$ 0.00, and Total: \$ 0.00.

In the workflow, the user should now have received the vendor bill either electronically, by mail, or in another form that they need to pay. Use the **Vendor** drop-down menu to select the vendor.

3. Next, use the **Auto-Complete** drop-down menu to select the purchase order named **PO00001 : \$ 87.00** as the one we wish to pay:

Draft Bill - First Number:
BILL/2019/0001

Vendor	T-shirt Supply Co.	Source Document	PO00001		
Vendor Reference		Bill Date			
Auto-Complete	Select a purchase order or an old bill	Due Date	06/03/2019		
		Bank Account			
<input type="radio"/> Bill <input type="radio"/> Other Info					
Product	Description	Quantity	Unit Price	Taxes	Amount
Medium White T-Shirt	PO00004: Medium White T-Shirt	12.000	7.25	<input type="checkbox"/> Tax 15.00%	\$ 87.00
Add a line					
Tax Description Tax 15.00%		Tax Amount \$ 13.05	Untaxed Amount: \$ 87.00 Tax: \$ 13.05 Total: \$ 100.05		
Add a line					

At this point, you will see all the necessary information from the purchase order, including the product, description, amount, and tax.

4. From here, you can click **Validate** to create the bill. This will then leave the bill in an open status until you register a payment or ask for a refund.
5. Now, let's pay this bill. Click on **Register Payment**:

Register Payment

Payment Amount	\$100.05	Payment Date	06/03/2019
Payment Journal	Bank (USD)	Memo	BILL/2019/0001
<input type="button" value="Validate"/> <input type="button" value="Cancel"/>			

Most of the information will automatically be filled out including **Payment Date** and **Payment Amount**. You can change these if necessary. The one piece of information you must provide is the **Payment Method**. For our example, we have chosen **Bank (USD)**.

The **Memo** can be used to create a note, should you wish to document this payment. For example, you can use the **Memo** field to specify the check number you use to pay your vendor.



When implementing a purchase order system, it is critical to train users thoroughly on how transactions are tied together. While many forms allow you to click on a link to view a related record, fields such as **Memo** store the reference to the other document just as text. Train and encourage users to quickly use copy and paste, rather than re-entering data into search fields.

- Odoo allows you to configure multiple payment methods. For now, choose **Bank** and click **Validate** to complete the transaction. You have now completed the entire purchasing cycle, from purchasing, to receiving, to finally paying for the product:

Product	Description	Quantity	Unit Price	Taxes	Amount
Medium White T-Shirt	PO00001: Medium White T-Shirt	12.000	7.25	Tax 15.00%	\$ 87.00

Tax Description	Tax Amount
Tax 15.00%	\$ 13.05

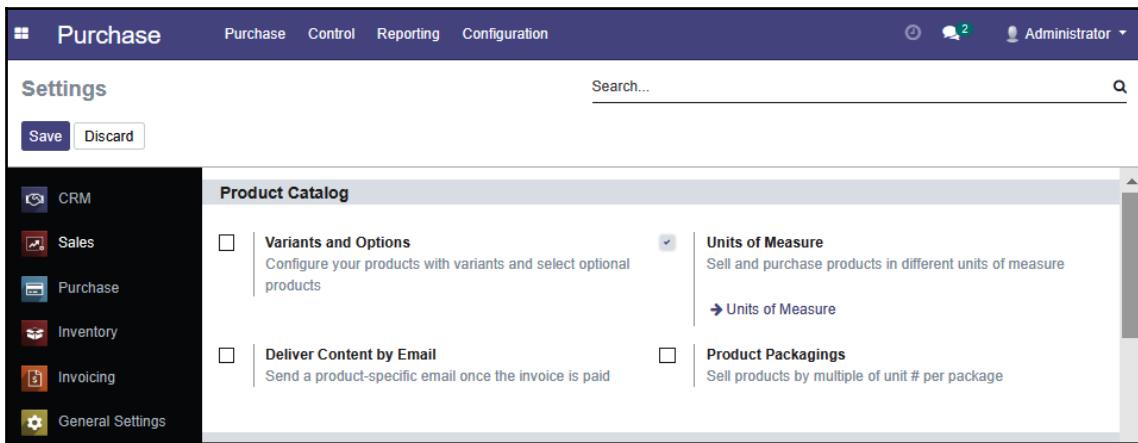
Untaxed Amount: \$ 87.00
Tax: \$ 13.05
Total: \$ 100.05
Paid on 06/03/2019 \$ 100.05
Amount Due: \$ 0.00

In the previous screenshot, you can see the final paid invoice for our purchased products. Notice how, at the bottom right of the form, it shows that we have a balance of zero. Now, we can also see a summary of the payment by hovering over the little information icon above the **Amount Due**.

Handling complex units of measure

By default, all the products you enter in to Odoo use the same unit of measure, units. Often, however, you might need to purchase and sell products in an alternate unit of measure. **Silkworm** has some products they purchase that come in cases. For example, they may need to purchase cases of coffee mugs they will sell as individual units in their retail stores. Let's see how we can configure Odoo to handle complex units of measure.

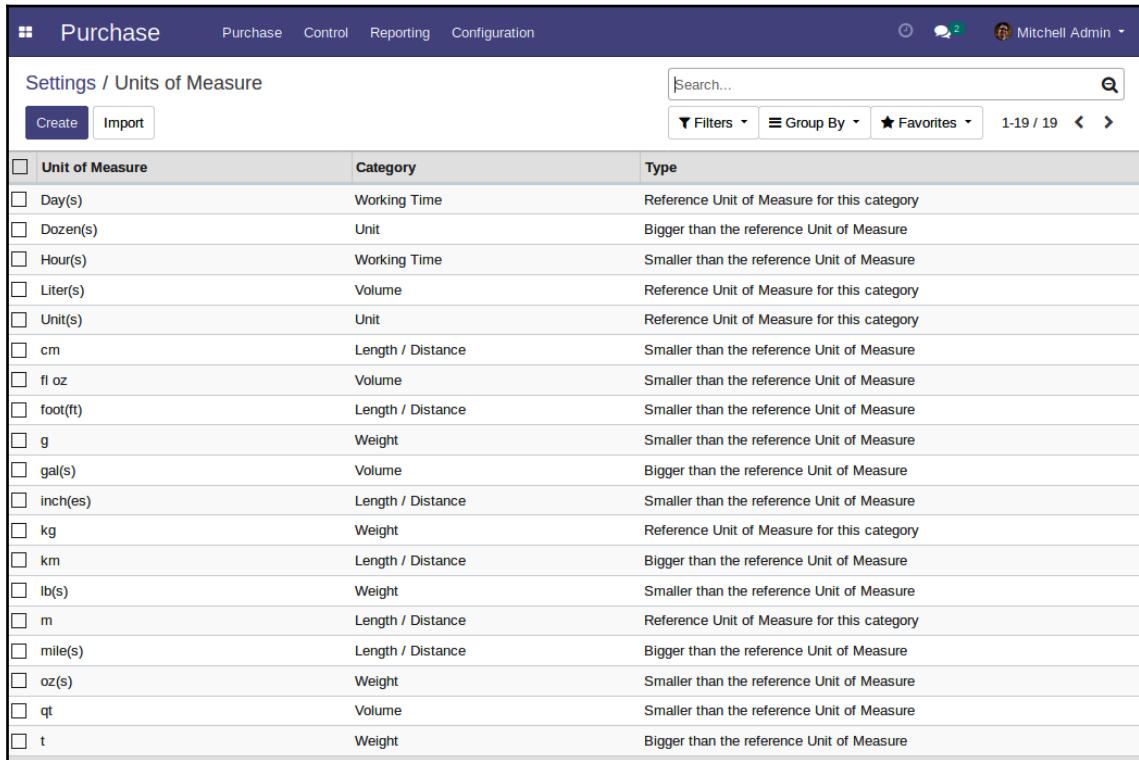
You turn on complex unit of measures by going to the **Sales** menu, then under **Configuration**, choosing **Settings**:



On the far right, you have the **Units of Measure** option. Check this option to allow multiple units of measure in Odoo. Click **Save** to confirm the change.

Looking at the defined units of measure

Odoo provides you with many predefined units of measure, including some of the most common ones. Click **Units of Measure** to see a list of the units of measure you can use for your products:



The screenshot shows the Odoo Purchase module interface. At the top, there's a navigation bar with 'Purchase' and sub-options: Purchase, Control, Reporting, Configuration. On the right, there are user icons and 'Mitchell Admin'. Below the navigation is a search bar and filter buttons for 'Filters', 'Group By', and 'Favorites'. A page number '1-19 / 19' and navigation arrows are also present. The main area is titled 'Settings / Units of Measure' with 'Create' and 'Import' buttons. A table lists various units of measure with their categories and types:

<input type="checkbox"/> Unit of Measure	Category	Type
<input type="checkbox"/> Day(s)	Working Time	Reference Unit of Measure for this category
<input type="checkbox"/> Dozen(s)	Unit	Bigger than the reference Unit of Measure
<input type="checkbox"/> Hour(s)	Working Time	Smaller than the reference Unit of Measure
<input type="checkbox"/> Liter(s)	Volume	Reference Unit of Measure for this category
<input type="checkbox"/> Unit(s)	Unit	Reference Unit of Measure for this category
<input type="checkbox"/> cm	Length / Distance	Smaller than the reference Unit of Measure
<input type="checkbox"/> fl oz	Volume	Smaller than the reference Unit of Measure
<input type="checkbox"/> foot(ft)	Length / Distance	Smaller than the reference Unit of Measure
<input type="checkbox"/> g	Weight	Smaller than the reference Unit of Measure
<input type="checkbox"/> gal(s)	Volume	Bigger than the reference Unit of Measure
<input type="checkbox"/> inch(es)	Length / Distance	Smaller than the reference Unit of Measure
<input type="checkbox"/> kg	Weight	Reference Unit of Measure for this category
<input type="checkbox"/> km	Length / Distance	Bigger than the reference Unit of Measure
<input type="checkbox"/> lb(s)	Weight	Smaller than the reference Unit of Measure
<input type="checkbox"/> m	Length / Distance	Reference Unit of Measure for this category
<input type="checkbox"/> mile(s)	Length / Distance	Bigger than the reference Unit of Measure
<input type="checkbox"/> oz(s)	Weight	Smaller than the reference Unit of Measure
<input type="checkbox"/> qt	Volume	Smaller than the reference Unit of Measure
<input type="checkbox"/> t	Weight	Bigger than the reference Unit of Measure

For our real-world example, let's assume we are going to purchase in **Dozen(s)**. This means that if we purchase one unit of a product that has a unit of measure of **Dozen(s)**, you will receive 12 individual units. If you purchase two dozens, you will receive 24 individual units. You can then choose to sell the product, either by **Dozen(s)** or by individual units.

Examining a specific Unit of Measure

Let's examine the unit of measure configuration for **Dozen(s)** by clicking it:

The screenshot shows the Odoo Purchase module interface. At the top, there is a navigation bar with tabs: Purchase, Control, Reporting, Configuration, and a user icon for Mitchell Admin. Below the navigation bar, the page title is "Settings / Units of Measure / Dozen(s)". There are two buttons: "Edit" (highlighted in blue) and "Create". To the right of these buttons is an "Action" dropdown and a page number "2 / 19" with navigation arrows. The main content area displays the configuration details for the "Dozen(s)" unit of measure. The fields are as follows:

Unit of Measure	Dozen(s)	Active	<input checked="" type="checkbox"/>
Category	Unit	Rounding Precision	0.01000
Type	Bigger than the reference Unit of Measure		
Ratio	12.00000		

Below the table, a note says: "e.g. 1 * (this unit) = ratio * (reference unit)".

At the top, we can see the name of our unit of measure, **Dozen(s)**. Odoo performs the calculations that are based on a reference unit of measure. In this case, the reference unit of measure is **Unit**. The **Type** is used to specify whether the **Unit of Measure** is smaller or bigger than the reference. Because **Dozen(s)** is the larger unit, we choose **Bigger than the reference Unit of Measure** for our calculations. The **Ratio** is then specified as 12 because there are 12 reference units to a dozen.



Examine existing **Units of Measure** and their relationships to help you better understand how to create your own custom configurations.

Configuring a product to be purchased in dozens

Click Products under Purchase and Create a new product.

In the following screenshot, you can see that we have set up a product, **Coffee Mug**. Most importantly, in this example, we have left our **Unit of Measure** as **Unit(s)**, but we have changed our **Purchase Unit of Measure** to **Dozen(s)**:

The screenshot shows the Odoo Purchase module interface. At the top, there are tabs for Purchase, Control, Reporting, and Configuration, with Purchase selected. On the right, there are notifications for 2 messages and a user named Administrator. Below the tabs, it says 'Products / New' with 'Save' and 'Discard' buttons. The main form is for creating a new product:

- Product Name:** Coffee Mug (highlighted in purple)
- Can be Sold:**
- Can be Purchased:**
- General Information:** Active
- Sales:** Sales Price: 22.00, Customer Taxes: Tax 15.00% (highlighted in purple)
- Purchase:** Cost: 0.00, Unit of Measure: Dozen(s) (highlighted in purple), Purchase Unit of Measure: Dozen(s) (highlighted in purple)
- Internal Notes:** A note: "This note is only for internal purposes."
- Bottom buttons:** Send message, Log note, Schedule activity, Follow, and a user icon.

With this configuration, when you issue a purchase order, you will specify the quantities of your order by the dozen. When you sell the product, you specify the quantity in individual units.

Summary

In this chapter, we installed the Purchase application and set up a vendor to purchase products. Then, we successfully purchased products and received those products. In Odoo 12, Inventory is no longer included during the installation of Purchasing; therefore, we saw how a more simplified process for receiving goods is managed in Odoo 12. After our products were received, we proceeded to pay the vendor bill in order to complete the payment cycle.

In the following chapter, *Chapter 5, Making Goods with Manufacturing*, we will take the raw materials we have just received in our inventory, and use them to manufacture and deliver a finished product. We will create manufacturing orders to define the steps of the production process and allocate the required resources. Coordinating all of your resources, including machinery and manpower, can be a daunting and time-consuming task, but we will learn how Odoo makes this process significantly more manageable.

5

Making Goods with Manufacturing

In this chapter, we will cover how you can use Odoo to manage the process of manufacturing products. Once you have received the required raw materials in your inventory, you can begin manufacturing the end product. Part of the functionality of an ERP system is to assist you in scheduling orders based on the available resources. One of these resources is, of course, the raw materials. Other resources could include the available labor or the availability of a particular machine. Essentially, the goal is to schedule the manufacturing order such that all the necessary resources are available and to produce the final product for on-time delivery.

In this chapter, we will cover the following topics:

- Making manufacturing orders
- Producing a custom product
- Building a work center
- Routing orders
- Creating orders with routing and a work center
- Reversing a bill of materials

Creating manufacturing orders

Manufacturing orders define the product you wish to build, the resources that are required, and the date by which you wish to produce the product. These resources often include raw materials that are sourced using a bill of materials. A bill of materials is essentially a list of products and their quantities that are required to produce the final product.

Producing the product

When it's time to produce the product, you then inform Odoo of each of the products that need to be produced, and your manufacturing order changes to a status of **Complete**. In a typical workflow, your raw materials are moved out of the inventory, and your finished product is added to your inventory.

Delivering the order

After a product has been produced and has been put into the inventory, it can be packaged and delivered to the customer. Depending on the specific manufacturing environment, a product may not even sit in a physical inventory location at all, and instead may be shipped almost immediately to the customer. Meanwhile, in another industry, you may have a product that is produced and then sits in a warehouse for months before delivery. Of course, it is always possible that something gets produced and gets left in dead stock. In this case, you would never have a delivery order and would instead use a process to determine how to manage that dead inventory.

Defining the workflow for your business

Much like configuring the CRM application, often the most complex part of setting up a purchasing and manufacturing system is not the ERP software itself. Instead, the real challenge is to understand the business requirements and how the current processes can be implemented in the best way possible. If you have never set up a purchasing and manufacturing system before, it is highly recommended that you supplement your knowledge with additional source material on the subject. Please refer to the Appendix A, *Locating Additional Odoo Resources* for references to additional resources on ERP and manufacturing.

A real-world example of a custom product

In Odoo, you manufacture products by creating manufacturing orders. For our example, we will be printing T-shirts that have a custom-designed logo. The basic manufacturing process involves using a screen to apply ink to each of the T-shirts. For now, we don't need to know all the details of this process to begin using Odoo to help schedule and track the manufacturing process of the product.

The basic steps in the process are simple:

1. Define a bill of materials that determines what items are needed to produce the final product.
2. Use a manufacturing order to print a design on the blank T-shirts.
3. Deliver the printed T-shirts to a customer.

Installing Manufacturing

Now, we must install the **Manufacturing** application so that we can begin configuring our T-shirt production. By now, you should be able to understand the modular nature of Odoo. Install the **Manufacturing** application just like you did with the other Odoo applications.

Choose **Apps** and install the **Manufacturing** application:



Clicking the **Install** button installs the **Manufacturing** application.



This **Manufacturing** application was formerly known as **Manufacturing Resource Planning (MRP)** in prior versions of Odoo.

Creating your first manufacturing order

The flexibility of Odoo provides a variety of approaches that you can take in order to set up your system. Manufacturing can also become a complex topic and is one of the more challenging aspects of setting up any ERP system. For our first manufacturing order, we will ignore many advanced options.



Keep it simple at first. There are many options available, and it will take some time for you to understand all of them. If you are new to manufacturing systems, it will take you longer to implement Odoo, and you should consider hiring professional consultants to assist you.

To create your first manufacturing order, go to the **Manufacturing** menu, choose **Manufacturing Orders**, and then click **Create**:

The screenshot shows the Odoo Manufacturing Orders / New window. At the top, there are buttons for Save and Discard. Below that is a toolbar with status indicators (Confirmed, In Progress, Done) and user information (Administrator). The main form has fields for Product, Quantity To Produce (set to 0.000), Bill of Material, Deadline Start (set to 06/25/2019 00:37:54), Responsible (set to Administrator), and Source. Below these fields are two tabs: Consumed Materials and Finished Products. A table below the tabs lists products with columns for Product, To Consume, Reserved, and Consumed. The table currently contains three empty rows.

This is the manufacturing order as it appears just after you hit **Create**.

Take a minute to look through the various fields and tabs to get a good idea of the information that is collected for a manufacturing order. Don't worry if you don't understand all the options yet. We will begin with a simple product and look at some of the most important aspects of creating a manufacturing order. Later in this chapter, we will explore some of the more complex manufacturing scenarios.

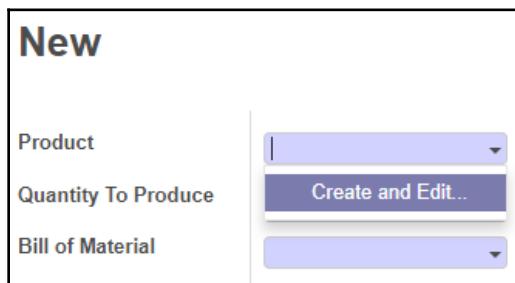
What product are we going to manufacture?

The only product we have entered into Odoo so far is Medium White T-shirt, which is blank. Nothing is printed on the T-shirt when it is received from the supplier. Now, we want to create a manufacturing order that will produce a new product in which we print a design of the customer's choice onto the T-shirt.

For our operations, we can still use the medium white T-shirt. But now, instead of selling the blank T-shirt directly to our customer, it will be used as a raw material for our manufacturing order.

Let's configure the final product that is to be created during the manufacturing process, that is, the complete T-shirt with the final design that will ship to the customer. For our example, this will be Class of 2017 T-shirt.

Odoo gives you the ability (in most forms where it is appropriate) to create a product on the fly. Let's create a product by clicking the drop-down arrow in the **Product** tab and choosing **Create and Edit...:**



Using the quick **Create and Edit...** option, you can add your finished products directly when creating a manufacturing order. In some workflows, where you may use a separate system to handle sales orders, this option can be a fast way to create the required finished products that you can then push into the inventory.

Next, you will fill out the **Product** form with the fields that are required for a finished product. On the **General Information** form, we want to make sure that we designate **Product Type** as **Storable Product**. This is important as it tells the Odoo **Manufacturing** application that we wish the final product to be stored in **Inventory**:

Create: Product

General Information

Product Name: Class of 2017 T-Shirt

Can be Sold
 Can be Purchased

0... Sold	Product Moves	0 Bill of Materials
0.00 Manufactured	0.00 Purchased	Active
0.00 On Hand	0.00 Forecasted	0 Reordering R...

Sales

Product Type: Storable Product

Sales Price: 25.00

Customer Taxes: Tax 15.00% *

Purchase

Cost: 12.00

Unit of Measure: Unit(s)

Inventory

Internal Notes:
This note is only for internal purposes.

Save **Discard**

In addition to making sure that you have **Product Type** specified, you can also provide **Sales Price** and **Cost** so that Odoo can provide proper reporting for inventory valuation and costing reports. Next, you will want to jump to the **Inventory** tab so that you can select the **Manufacture** route. You will also want to uncheck the **Buy** option under **Routes** as this is not a product that we can purchase. We must make it part of our manufacturing operation.

The **General Information** and **Inventory** tabs should look familiar by now. Starting in Odoo 9, Odoo provides routes, which allow a lot of flexibility in defining complex workflows and supplying management operations. Fortunately, the more common routes are already configured, and we can check **Manufacture** to let Odoo Manufacturing know that this product is part of the manufacturing workflow. Only products that have the supply method of **Manufacture** can be selected as a product on a manufacture order.

Also, note at the top of the form that the **Can be Purchased** box is unchecked. This will keep this product from appearing in the product list on a purchase order. Since we cannot purchase this product directly from a supplier, we don't want it appearing in our product list when we're working within purchasing.

Building your Bill of Materials

A **Bill of Materials (BoM)** is essentially a list of products that are required to produce another product. You can think of it like the list of ingredients for a recipe. Odoo needs to know what materials are required for us to produce this **Class of 2017 T-shirt** product.

In complex products, a BoM can be nested. For example, it may take many products to make a subproduct, and then several subproducts to make a final product.



Don't let nested BoMs intimidate you. Once you understand how a simple BoM is processed, you will see how you can group parts together. Think about grouping more complex BoMs by assembly and work centers. This makes it easier to see your inventory in real time since BoMs can be processed at each stage of your operations, properly using up materials and creating finished sub-assemblies.

For our first BoM example, we will be keeping it simple. We are just going to require the white T-shirt. The rest of the operation, that is, printing the actual T-shirt, will be incorporated into the manufacturing order. In other words, if there are enough white T-shirts, this manufacturing order can be processed, and we can produce the final product. For now, the inks and screens will not be managed in the manufacturing process. This is an example of starting simple and adding more complexity as we build up the system.

A smart button at the top right of the form shows you a count of BoMs that are attached to this product:



Clicking this button will bring up the **Bill of Materials** listing for that product:

The screenshot shows the Odoo Manufacturing interface. At the top, there's a navigation bar with tabs for Manufacturing, Operations, Master Data, Reporting, and Configuration. On the far right of the bar, there are icons for a user profile (Administrator), notifications (5), and a search function. Below the navigation bar, the main area has a breadcrumb trail: Products / Class of 2017 T-Shirt / Bill of Materials. To the left of the trail are two buttons: 'Create' and 'Import'. To the right is a search bar with the placeholder 'Search...' and a magnifying glass icon. Below the search bar are three dropdown buttons: 'Filters', 'Group By', and 'Favorites'. The central part of the screen is a large, empty white area. In the upper-left quadrant of this area, there's a circular icon containing a white document with a smiley face and some horizontal lines, resembling a notepad. Below this icon, the text 'Create a new document' is displayed.

Naturally, this is a blank list since we haven't defined any BoMs for this product yet. In the top-right corner, the product filter is restricting this list view to only display the **Bill of Materials** that is for the Class of 2017 T-Shirt.

Clicking **Create** will now bring up a blank **Bill of Materials** form with Class of 2017 T-Shirt automatically prepopulated as the finished product to build.

Many of the fields will be automatically filled out as Odoo knows we are creating a BoM for our Class of 2017 T-Shirt product. In this example, we have added Medium White T-Shirt to the **Bill of Materials**. When we manufacture one Class of 2017 T-Shirt, we will require one medium white T-shirt:

The screenshot shows the Odoo Manufacturing Bill of Materials (BoM) screen. At the top, there are tabs for Operations, Master Data, Reporting, and Configuration. On the right, there are icons for Help, Notifications (5), and Administrator. The main title is "Products / Class of 2017 T-Shirt / Bill of Materials / New". Below the title are "Save" and "Discard" buttons. The main area has sections for Product (set to "Class of 2017 T-Shirt"), Quantity (set to "1.00 Unit(s)"), and BoM Type (set to "Manufacture this product"). There are also "Active" and "Structure & Cost" buttons. A "Components" tab is selected, showing a table with one row: "Medium White T-Shirt" with a quantity of "1.000 Unit(s)". Below the table is a link "Add a line". At the bottom, there are buttons for "Send message" and "Log note", and a footer with "Follow" and "0" notifications.

Often, if not most of the time, a BoM will contain multiple items. Regardless of the number of items in the BoM, the way they are processed is the same.



It is possible that your manufacturing order or BoM screen may look slightly different to the ones you can see here. One reason for this is that, depending on the modules that are installed and what options are selected, the forms may have different content. Another common reason is that Odoo is currently getting frequent updates that can change the appearance of any given form.

Setting a product as storable

By default, Odoo sets stock up as being consumable. When we add products to a BoM, we typically want that product to be managed in the inventory. If we processed our manufacturing order with Medium White T-Shirt set to consumable, then it would not check for the availability of the product and assume that it is available. This isn't the typical behavior we would want in our manufacturing system. Open the Medium White T-Shirt product using the small pop-out icon on the right:



Here you can edit the **Medium White T-Shirt** product as a component of the BoM for our Class of 2017 T-Shirt:

Open: Component

Update Qty On Hand
Replenish

Product Name

Medium White T-Shirt

0.00 Unit(s) Sold	12.000 Unit(s) On Hand	11.000 Unit(s) Forecasted
Product Moves	0 Reordering R...	0 Bill of Materials
More ▾		

General Information
Sales
Purchase
Inventory

Product Type: Storable Product

Internal Reference:

Barcode:

Product Category: All

Sales Price: \$16.50

Customer Taxes: Tax 15.00% ×

Cost: \$7.25

Unit of Measure:

Purchase Unit of Measure:

Unit(s):

Unit(s):

Internal Notes

This note is only for internal purposes.

Save
Discard

Now, set **Product Type** to **Storable Product**. With the product configured this way, Odoo manufacturing will check to make sure that the product is in the inventory.

Confirming production

Now, we can open our **Manufacturing Orders** and begin production. Once you click on **Confirm Production**, you are ready to manufacture the product. Odoo will provide reasonable defaults, which you can override as required. When production is confirmed, this doesn't mean that production has taken place. Confirming production has only informed the system that production is ready to proceed. You can tell we are ready to begin manufacturing because the **Produce** button is available.

Here is what our Class of 2017 T-Shirt in **Manufacturing Orders** looks like now:

The screenshot shows the Odoo Manufacturing Orders interface. At the top, there's a navigation bar with tabs for Manufacturing, Operations, Master Data, Reporting, Configuration, and a user icon for Administrator. Below the navigation bar, the main title is "Manufacturing Orders / WH/MO/00001". There are buttons for Edit, Create, Print, Action, and a status indicator showing 1/1. A toolbar below the title includes buttons for Check availability, Produce, Cancel, Scrap, and Unlock, along with a status progress bar showing Confirmed, In Progress, and Done. The main content area displays the details of the production order WH/MO/00001. It shows the Product as "Class of 2017 T-shirt", Quantity To Produce as "1.000 Unit(s)", and Bill of Material as "Class of 2017 T-shirt". The Deadline Start is listed as 06/25/2019 01:19:49, Responsible is "Administrator", and Source is " ". Below this, there are tabs for Consumed Materials and Finished Products. Under Consumed Materials, there's a table with one row for "Medium White T-Shirt" with a unit of measure of "Unit(s)". The columns are Product, Unit of Measure, To Consume, Reserved, and Consumed. The values are 1.000, 1.000, and 0.000 respectively. At the bottom of the screen, there are buttons for Send message, Log note, Schedule activity, and social sharing options (Following, 1 follower).

This is the manufacturing form showing the products waiting to be consumed to produce the order. In this case, it is our **Medium White T-Shirt**.

Checking availability

Odoo Manufacturing links into the inventory automatically, and it will use available stock to complete the order. Since we already purchased **Medium White T-Shirts** in the previous chapter, clicking the **Check Availability** button will hide the button and remove the red highlight on **Medium White T-Shirt**, as well as remove the material warning message.

Please be aware that, at this stage, if you don't have an available quantity, Odoo will continue to display the **Check Availability** button and the warning message until the product is acquired and put into the inventory. Each time you click **Check Availability**, Odoo will look in the inventory to see whether we have the necessary products to complete the manufacturing order. Alternatively, you can click **Produce** to tell Odoo Manufacturing that you are ready to produce this product, even if the inventory within the system doesn't meet the necessary requirements. Note that doing this will give you negative inventory quantities in your warehouse.



Using what you have learned in *Chapter 4, Purchasing with Odoo*, you can create a purchase order to purchase products and put them into the inventory. Just make sure that you have set **Product Type to Storable Product**.

After the raw product has been acquired and we are ready to produce our final product, the form will be updated and the state will be changed:

Manufacturing Orders / MO/00001

Edit Create Print Action 1 / 1 < >

Produce Cancel Scrap Unreserve Unlock Confirmed In Progress Done

MO/00001

Product	Class of 2017 T-Shirt	Deadline Start	11/14/2017 15:39:55
Quantity To Produce	1.000 Update	Responsible	Administrator
Bill of Material	Class of 2017 T-Shirt	Source	

Consumed Materials Finished Products Miscellaneous

Product	To Consume	Reserved	Consumed
Medium White T-Shirt	1.000	1.000	0.000

Here, we can see the quantity of medium white T-shirts available, the amount we will consume in the manufacturing order, as well as the total that will be consumed once the order is complete.

Producing the product

After you click on the **Produce** button, you will be prompted to confirm that the product has been produced:

Produce

Product	Class of 2017 T-shirt
Quantity	1.000 Unit(s)

Product	To Consume	Reserved	Consumed	Unit of Measure
Medium White T-Shirt	1.000	0.00	1.00	Unit(s)

Add a line

Record Production Cancel

For our example, we will leave the quantity set to one unit.

Click the **Record Production** button to produce the product:

The screenshot shows the Odoo Manufacturing Orders interface. At the top, there are tabs for Operations, Master Data, Reporting, and Configuration. On the right, there are icons for search, administrator, and a dropdown menu. Below the tabs, it says "Manufacturing Orders / WH/MO/00001". There are buttons for Edit, Create, Print, Action, Mark as Done, Check availability, Scrap, Unlock, Confirmed, In Progress (which is highlighted), and Done. The main area displays manufacturing order details: Product (Class of 2017 T-shirt), Quantity To Produce (1.000 Unit(s)), Bill of Material (Class of 2017 T-shirt), Deadline Start (06/25/2019 01:19:49), Responsible (Administrator), and Source. A button labeled "Inventory Moves" with a double arrow icon is visible. Below this, there are tabs for Consumed Materials and Finished Products, with Consumed Materials selected. A table shows the consumed materials: Medium White T-Shirt (Unit(s)) with To Consume (1.000), Reserved (0.000), and Consumed (1.000). At the bottom, there are buttons for Send message, Log note, Schedule activity, and a Following status indicator.

The product has been produced and is ready to be posted into **Inventory**. If you go and look at the **Product** record now, you will not see any **Class of 2017 T-shirt** at hand. At this stage, you can click the **Mark as Done** button and Odoo will set both **Inventory** and **Mark the Manufacturing Order** as done. Alternatively, you could click the **Post Inventory** button to add **Class of 2017 T-Shirt** to **Inventory**, and then use a separate step to click the **Mark as Done** button to move the order to the **Done** state.

Congratulations, you have just used Odoo to manufacture your first product!

Analyzing stock valuation

In our example, we have taken a raw material and increased its value by producing a finished product. One of the easiest ways to see the effect of our manufacturing order is to look at the **Inventory Valuation** report, which can be found by going to **Inventory** | **Reporting** | **Inventory Valuation**.

Here, you will see that we now have 1 unit of **Class of 2017** T-shirt and 11 units of **Medium White T-Shirt**:

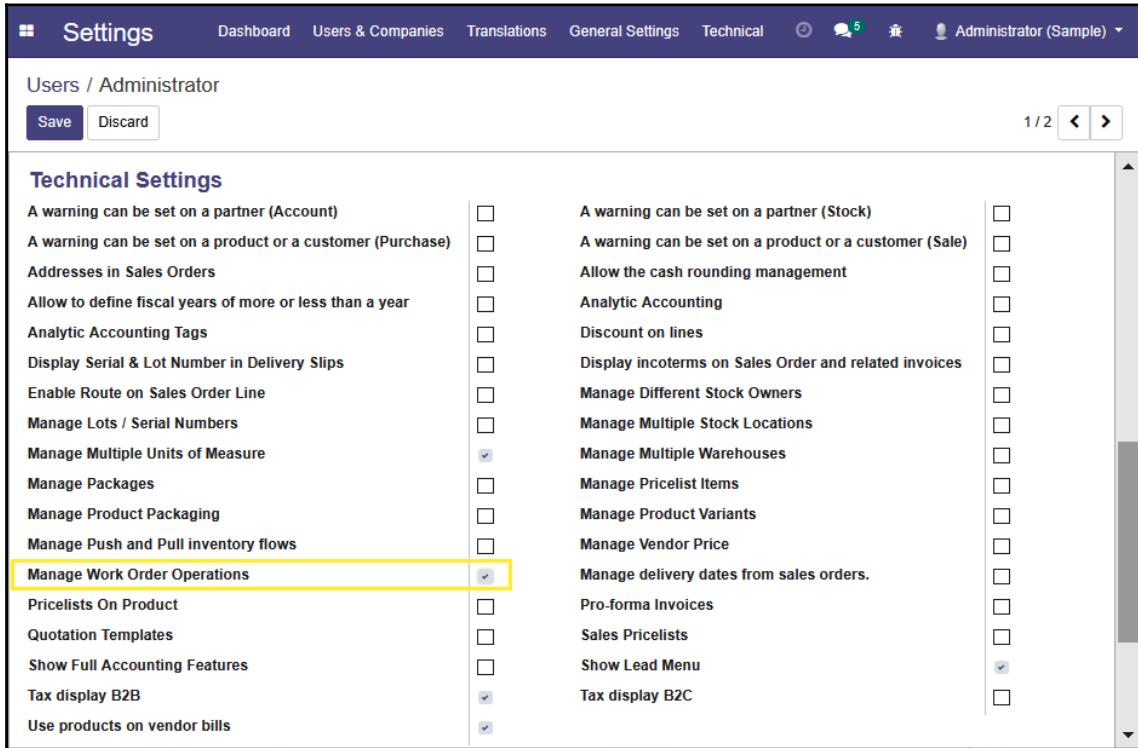
<input type="checkbox"/> Display Name	Quantity	Unit of Measure	Value
<input type="checkbox"/> Class of 2017	1.00	Unit(s)	\$ 12.00
<input type="checkbox"/> Medium White T-Shirt	11.00	Unit(s)	\$ 79.75
91.75			

The inventory is accurately reflecting the purchases that have been made by us, as well as the products that have been consumed and produced by our manufacturing order. The value of our inventory went from **\$87.00** to **\$91.75** merely by printing a single T-shirt.

Managing production by work orders

This first manufacturing order was very simple, and our BoM only contained one product. In many companies, the manufacturing operations are far more complex. For example, in some instances, depending on the attributes of the product, the manufacturing could involve different work centers or alternative steps to produce the final product. By default, Odoo's manufacturing application takes a more simplified approach.

In Odoo 12, you must turn on developer mode by going to **Settings** and then clicking the **Activate the Developer Mode** link on the far right. Now, you can go to the user that needs to manage the work orders and select the appropriate option:



The screenshot shows the Odoo 12 Settings page under the Technical tab. The 'Technical Settings' section contains various configuration options, each with a checkbox. The 'Manage Work Order Operations' checkbox is highlighted with a yellow box. Other visible options include: A warning can be set on a partner (Account), A warning can be set on a product or a customer (Purchase), Addresses in Sales Orders, Allow to define fiscal years of more or less than a year, Analytic Accounting Tags, Display Serial & Lot Number in Delivery Slips, Enable Route on Sales Order Line, Manage Lots / Serial Numbers, Manage Multiple Units of Measure (with a checked checkbox), Manage Packages, Manage Product Packaging, Manage Push and Pull inventory flows, Manage Work Order Operations (with a checked checkbox), Pricelists On Product, Quotation Templates, Show Full Accounting Features, Tax display B2B, Use products on vendor bills, A warning can be set on a partner (Stock), A warning can be set on a product or a customer (Sale), Allow the cash rounding management, Analytic Accounting, Discount on lines, Display incoterms on Sales Order and related invoices, Manage Different Stock Owners, Manage Multiple Stock Locations, Manage Multiple Warehouses, Manage Pricelist Items, Manage Product Variants, Manage Vendor Price, Manage delivery dates from sales orders, Pro-forma Invoices, Sales Pricelists, Show Lead Menu, and Tax display B2C.

After you apply the changes, the menus will refresh, and new options will be added to the **Manufacturing** application.

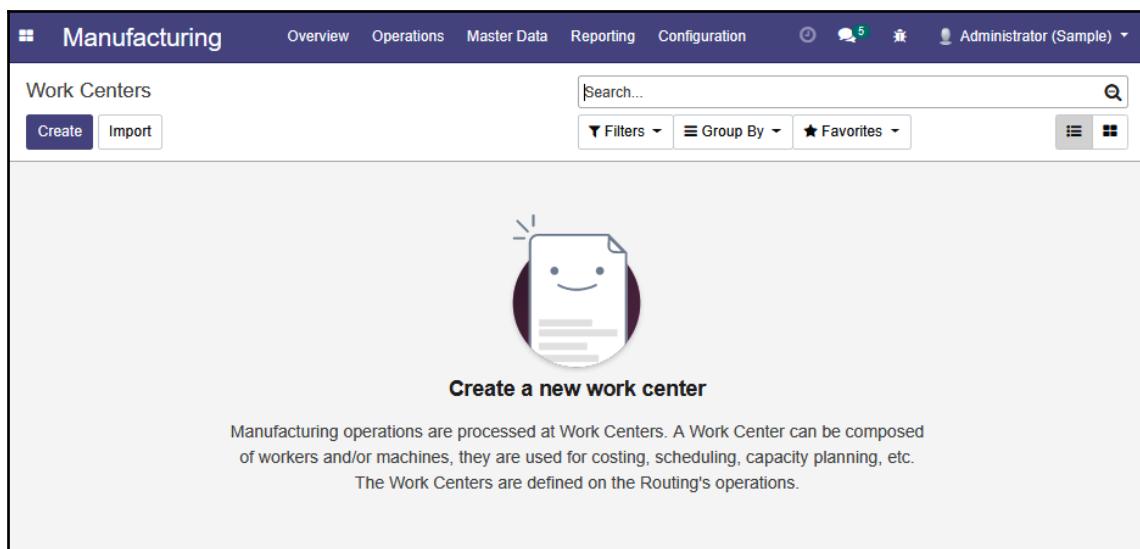


Sometimes, when adding new functionality to Odoo, such as when adding applications or modifying settings, it can be helpful to do a *Shift*-refresh on your browser to make sure that Odoo is refreshed with the latest options.

Creating a work center

In our previous simplified manufacturing order, we specified the raw product that's required in a BoM, and then turned that into a finished product. Now, we will expand this example to specify the human labor that goes into printing our **Class of 2017 T-Shirt**. In Odoo, we define parameters in a work center.

For our example, we will create a work center called **Printing** that is responsible for taking the blank T-shirt and applying the design to create the final product. We will begin by going to the **Manufacturing** application, and, under the **Master Data** menu, choosing the **Work Centers** option:



The screenshot shows the Odoo Manufacturing application's Work Centers page. At the top, there is a navigation bar with tabs for Overview, Operations, Master Data, Reporting, Configuration, and a notification badge for 5 messages. The Master Data tab is selected. On the left, a sidebar shows a tree view of categories like Manufacturing, Sales, and Purchasing. The main content area is titled "Work Centers". It features a search bar with placeholder "Search..." and icons for Filters, Group By, and Favorites. Below the search bar are two buttons: "Create" (highlighted in blue) and "Import". A large central area contains a cartoon illustration of a smiling document with a pencil and a speech bubble. Below the illustration, the text "Create a new work center" is displayed. A descriptive text block follows: "Manufacturing operations are processed at Work Centers. A Work Center can be composed of workers and/or machines, they are used for costing, scheduling, capacity planning, etc. The Work Centers are defined on the Routing's operations." At the bottom right of the main area, there are two grid icons for "List View" and "Table View".

Like always, Odoo provides helpful instructions on creating a new work center. Click **Create** to set up a new work center record:

The screenshot shows the Odoo Manufacturing interface for creating a new work center. At the top, there's a navigation bar with tabs like Overview, Operations, Master Data, Reporting, Configuration, and a user dropdown for 'Administrator (Sample)'. Below the navigation is a sub-header 'Work Centers / New' with 'Save' and 'Discard' buttons.

The main form area has several sections:

- General Information:** Shows OEE metrics (0.00% OEE, 0.00 hour(s) Lost, 0.00 minute(s) Work Center ...), a performance bar (0%, Active), and fields for 'Work Center Name' (set to 'Printing'), 'Code' (empty), and 'Working Hours' (set to 'Standard 40 Hours/Week').
- Costing Information:** Includes fields for 'Time Efficiency' (100.00%), 'Capacity' (1.00), 'OEE Target' (90.00%), 'Cost per hour' (0.00), and time intervals before and after production ('Time before prod.' and 'Time after prod.' both at 00:00).
- Description:** A text area for describing the work center, with placeholder text 'Description of the work center...'. There's also a small icon of a person in the bottom right corner of this section.

In our example, we have named the work center **Printing**. In a full implementation, it would be common to have different work centers based on the work performed. There is also an optional field for **Code**. This could be a short description that you use internally to designate a work center. By default, **Working Hours** is set to **Standard 40 Hours/Week**.

Setting general information

When defining a work center, it is possible to define **General Information**, which will allow you to estimate the cost and time that's required to produce your products. In our example, we are going to configure this work center so that we can estimate the time that's required to produce a T-shirt.

Time Efficiency

Time Efficiency is a metric of how efficient this work center is at completing tasks. Often, the **Time Efficiency** is most valuable in allowing you to tweak your work center capacity without modifying any requirements on the manufacturing order. If, for example, you have an efficiency factor of 200.00 (or 200%), then the work center will complete twice as many tasks. For our example, we are leaving the efficiency factor as the default of 100.00 (or 100%).

Capacity

The **Capacity** metric allows you to determine how many tasks the work center can do in parallel. For example, if you have a work center that could be configured with three workers and all three workers can complete a cycle at the same time, you could set the capacity per cycle to three. When a manufacturing order is then routed to the work center, the work center can complete three tasks at the same time. For our example, we will assume there is one worker, and therefore one capacity per cycle.

Overall equipment effectiveness target

In Odoo 12, the Manufacturing application now places far more emphasis on tracking and reporting. A central metric to the information that's provided is **overall equipment effectiveness (OEE)**. This is essentially a calculation that determines what percentage of your manufacturing time is used in actual production. The closer this is to 100%, the more efficient use you are making of your resources. By setting a work center goal, you can determine how well your work center is meeting expectations.

Time before and after production

Many work center operations will have time required for setup and teardown outside of the time consumed by actual production. This is certainly true for our T-shirt printing example. It takes time for someone to prepare a printing press with inks before the first T-shirt can be printed. For our example, we have estimated five minutes of setup time. Likewise, when we are done producing the last product in our work order, it takes time to clean up before setting up the next job. In this example, we have estimated another five minutes of time at the end of production for the **Time after prod.** operations.

Creating routing orders

After defining a work center, you need to define a way to specify under which conditions you should use the work center. This is accomplished by defining routings. For our example, we are going to keep it simple and use routing to send our manufacture order to the printing work center for the finished product to be produced. In a real-world example, the job may use routings to go through many work center operations before the final product is produced.

To create a routing order, go to the **Manufacturing** application and choose **Routings** under the **Master Data** submenu. Click **Create** to bring up the new **Routing** form:

The screenshot shows the SAP Manufacturing application interface. At the top, there is a navigation bar with links for Overview, Operations, Master Data, Reporting, Configuration, and several user icons. Below the navigation bar, the title "Manufacturing" is displayed, followed by "Routings / New". There are two buttons: "Save" and "Discard". On the right side of the screen, there are two status indicators: "Time Analysis" (with a clock icon) and "Active" (with a document icon). The main area is titled "New" and contains a "Routing" section with a dropdown menu showing "Print Job" (which is highlighted with a blue background). Below this, there are tabs for "Work Center Operations" and "Notes". A table is present with the following data:

Operation	Work Center	Duration
Add a line		

In our example, we have named the routing `Print Job`.

Next, we will define our work station as **Operation** by clicking **Add a line** and bringing up the **Create Operations** form:

The screenshot shows the 'Create Operations' dialog box. The 'Operation' field is set to 'Print T-Shirt'. The 'Work Center' dropdown is set to 'Printing'. The 'Sequence' field contains '100'. Under 'Duration Computation', the 'Set duration manually' option is selected, with '60:00 minutes' entered. A description box contains the text 'This operation prints T-Shirts at the Printing Work Center|'. At the bottom, there are four buttons: 'Save & Close' (highlighted in purple), 'Save & New', 'Discard', and 'Remove'.

When defining our **Operation**, we can name it whatever we wish, but in this case, I picked **Print T-Shirt**. This indicates that this operation is more specific than the simple **Print Job** we are assigning to the routing order.

For complex routings, you can specify the sequence of the operations. We could, for example, have a **Design** operation and a **Build Screen** operation before the **Print Job** operation. Then, we could specify a **Quality Assurance** operation and a **Packing** operation after the **Print Job** operation. You would handle all of these exactly the same way as you set up the printing work center and created the required operations to produce the product. By starting simple and adding additional operations and complexity over time, you can often get up and running much more quickly than trying to track every little task right from the beginning.

Once you have set up your **Operation**, your routing should resemble the following form:

Operation	Work Center	Duration
Print T-Shirt	Printing	60:00

Here, we can see the finished routing, along with the operation details.

Assigning the routing to a BoM

Now that we have created a routing, we need to tell the **Bill of Materials** to use our newly defined **Routing**. In previous versions of Odoo, it was possible to assign a **Routing** right to the work order. In Odoo 12, if you don't set up the **Routing** on the **Bill of Materials**, it will not be available to you when you create a manufacturing order.

Pull up the **Bill of Materials** for our **Class of 2017 T-Shirt**:

Component	Quantity	Product Unit of Measure	Consumed in Operation
Medium White T-Shirt	1.00	Unit(s)	

Set the routing to the **Print Job** operation we just created, as shown in the preceding screenshot.

Creating a manufacturing order with routing and a work center

Now that we have defined our work center and our routing operation, we can create a manufacturing order that will utilize our new production steps. In this example, we are going to produce 15 units of **Class of 2017 T-Shirt**. Start by going to the **Operations** menu and choosing **Manufacturing Orders**. Then, click **Create**:

The screenshot shows the Odoo Manufacturing Orders interface. At the top, there's a navigation bar with tabs for Overview, Operations, Master Data, Reporting, and Configuration, and a user dropdown for Administrator. Below the navigation bar, the main title is "Manufacturing Orders / WH/MO/00002". There are buttons for Edit, Create, Print, Action, Check availability, Create Workorders, Cancel, Scrap, and Unlock. A progress bar at the top right indicates the order is at step 3/3, with status boxes for Confirmed, In Progress, and Done. The main content area displays the details of the manufacturing order:

Product	Class of 2017 T-Shirt	Deadline Start	06/25/2019 12:15:40
Quantity To Produce	15.000 Unit(s) Update	Responsible	Administrator
Bill of Material	Class of 2017 T-Shirt	Source	
Routing	Print Job		

Below this, there are tabs for Consumed Materials and Finished Products. Under Consumed Materials, there is a table:

Product	Unit of Measure	To Consume	Reserved	Consumed
Medium White T-Shirt	Unit(s)	15.000	0.000	0.000

When we select the **Product**, Odoo will automatically assign the associated BoM for the product. You will notice in the manufacturing order that Odoo automatically selected **Print Job** for the **Routing** of this order. This is the key field that will send this job to the printing work center to be produced.

When you click **Save**, you will notice that Odoo hasn't automatically created work orders. There is a separate button in Odoo 12 to trigger the creation of the work orders. Let's go over how to do this in the following steps:

1. Click the **Create Work Orders** button to create the work order.

You can now go to the **Work Orders** option under **Operations** to view the work order in the default Kanban view:

The screenshot shows the Odoo Manufacturing module's Kanban view for Work Orders. The top navigation bar includes 'Manufacturing', 'Overview', 'Operations', 'Master Data', 'Reporting', 'Configuration', and 'Administrator'. Below the header is a search bar with filters, group by, and favorite options. A single work order card is displayed, showing 'WH/MO/00002 - Print T-Shirt' and 'Class of 2017 T-Shirt 15.000 Unit(s)'. The card has a blue border.

2. Clicking on **Class of 2017 T-Shirt** would then bring up the work order for you to review:

The screenshot shows the Odoo Work Order details view for 'WH/MO/00002 - Class of 2017 T-Shirt - Print T-Shirt'. The top navigation bar is identical to the previous screenshot. The main area shows the work order details: 'Start Working' button, status bar with 'Pending', 'Ready', 'In Progress', and 'Finished' stages, and a 'Waiting Materials' label. It also displays 'To Produce' (Class of 2017 T-Shirt), 'Quantity Produced' (0.000 / 15.000 Unit(s)), and 'Current Qty' (15.000). Navigation buttons at the bottom indicate this is page 2 of 2.

As you can see from the red **Waiting Materials** label, we are warned that we lack the materials to produce the product.

3. Still, we can ignore this message and click the **Start Working** button at the top of the form to start producing the product. If you wish, you could use what you learned in Chapter 4, *Purchasing with Odoo*, to purchase enough raw products to get rid of the **Waiting Materials** label.
4. At this point, Odoo considers this to be a real-time tracking system. That means that when your worker begins the job, that is when they should click the button. When the workers have completed the job, they should click the **Done** button. By implementing tablets or other workstations that are capable of running Odoo, you can create a manufacturing environment that automatically tracks the time of your operations.

If you click on the **Time Tracking** tab, you can see details on the current operations, as well as a timer that displays the updated duration of the operation in real time:

The screenshot shows the Odoo Manufacturing module interface. At the top, there's a navigation bar with tabs: Overview, Operations, Master Data, Reporting, Configuration, and a user icon for Administrator. Below the navigation bar, the main title is "Work Orders / WH/MO/00002 - Class of 2017 T-Shirt - Print T-Shirt". There are four buttons in a row: Done (green), Pause (yellow), Block (red), and Scrap (grey). To the right of these buttons is a progress bar with steps: Pending, Ready, In Progress, and Finished. The progress bar is currently at the 'Pending' step. Below the buttons and progress bar, there's a section for "To Produce": "Class of 2017 T-Shirt" and "Quantity Produced: 0.000 / 15.000 Unit(s)". A red box highlights the "Waiting Materials" status. Under the "Time Tracking" tab, there's a table with columns: Start Date, End Date, Duration, User, and Productivity. The table has one row: Start Date 06/25/2019 14:13:42, End Date 06/25/2019 14:13:42, Duration 00:00, User Administrator, Productivity Fully Productive Time. A large grey bar at the bottom represents the timer, with the text "00:00" in the center.

5. Now that you have had a chance to see the operation running, let's go ahead and click **Done** so that we can see the results of our work in Odoo.

Odoo now refreshes the screen and moves the state of the workstation operation to **Done**. One thing that needs to be mentioned is that **Work Centers** can perform their operations to completion, regardless of the quantities in **Inventory**. This means that you would most likely have an internal process in which the worker would first check the availability on the manufacturing order before the worker even bothers to pull up the work order.

6. Now, let's take a look at the manufacturing order:

The screenshot shows the Odoo Manufacturing Orders interface. At the top, there are tabs for Overview, Operations, Master Data, Reporting, and Configuration. On the right, there are notifications for 2 messages and a user named Administrator. Below the tabs, it says "Manufacturing Orders / WH/MO/00002". There are buttons for Edit, Create, Print, Action, Mark as Done, Check availability, Scrap, Unlock, and a status indicator (3 / 5). A progress bar shows Confirmed, In Progress, and Done. The main area displays the details of Manufacturing Order WH/MO/00002. It includes fields for Product (Class of 2017 T-Shirt), Quantity To Produce (15.000 Unit(s)), Bill of Material (Class of 2017 T-Shirt), Routing (Print Job), Deadline Start (06/25/2019 12:15:40), Responsible (Administrator), and Source. To the right, there are labels for 1 / 1 Work Orders and Inventory Moves. Below this, there are two tabs: Consumed Materials and Finished Products. The Consumed Materials tab shows a table with one row for Medium White T-Shirt with a unit of measure of Unit(s), quantity of 15.000, reserved quantity of 0.000, and consumed quantity of 15.000. The Finished Products tab shows a table with one row for Medium White T-Shirt with a unit of measure of Unit(s), quantity of 15.000, reserved quantity of 0.000, and consumed quantity of 15.000.

In the top right, we can see from the **1/1 Work Orders** label that we have processed all the work orders for this manufacturing order. We can see that because that was the only work order required to complete the manufacturing order; the **Consumed Materials** tab shows that the raw products have been consumed. Furthermore, the **Finished Products** tab already shows the 15 T-shirts as being produced. The order will continue to stay as **In Progress** until you click the **Mark as Done** button. At that point, the **Manufacturing Orders** will simply drop out of the list.

7. Now, if you go to **Inventory Valuation**, you will see the results of these operations. Remember that although we didn't have enough inventory of the **Medium White T-shirts**, we still went ahead and completed the work order. Fortunately, Odoo keeps complete track of everything so that we can properly reconcile the inventory later. Therefore, we can see a negative number of **Medium White T-Shirts** in our inventory, as well as the 16 units of **Class of 2017 T-Shirt** that were produced from the two manufacturing orders in this chapter:

The screenshot shows the Odoo inventory management interface. At the top, there's a navigation bar with tabs for Overview, Operations, Master Data, Reporting, and Configuration. On the right side of the header, there are icons for Help, Chat, and Administrator, along with a user profile picture. Below the header, the main title is "Inventory Overview / Inventory Valuation". To the right of the title is a search bar with placeholder text "Search..." and a magnifying glass icon. Underneath the search bar are three dropdown menus: "Filters", "Group By", and "Favorites". To the right of these is a page number "1-2 / 2" and navigation arrows. The main content area is a table with the following data:

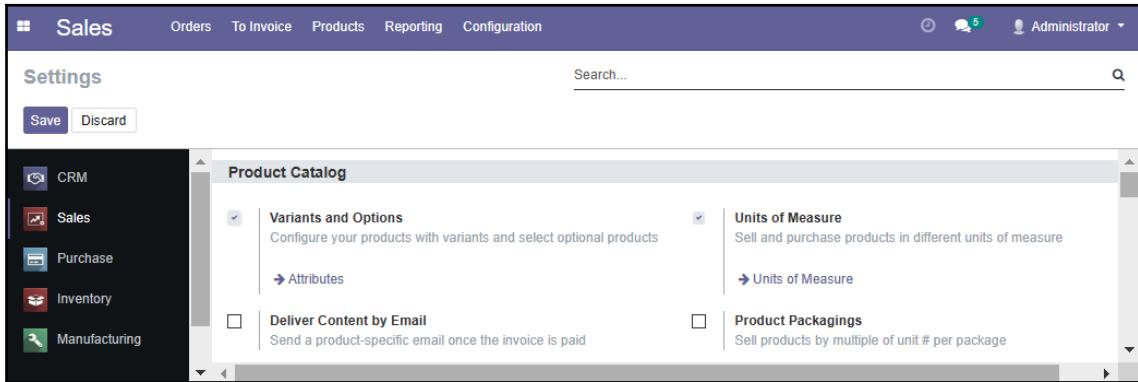
Display Name	Quantity	Unit of Measure	Value
Class of 2017 T-Shirt	16.00	Unit(s)	\$ 192.00
Medium White T-Shirt	-4.00	Unit(s)	\$ - 29.00
163.00			

8. We have also set the cost of our **Medium White T-Shirt** to **\$7.25 (per Unit)** and our **Class of 2017 T-Shirt** to **\$12.00 (per Unit)** inside their respective product records. This allows us to calculate the value of our inventory. Next, we click **Attributes** and set up the variants we wish to use.

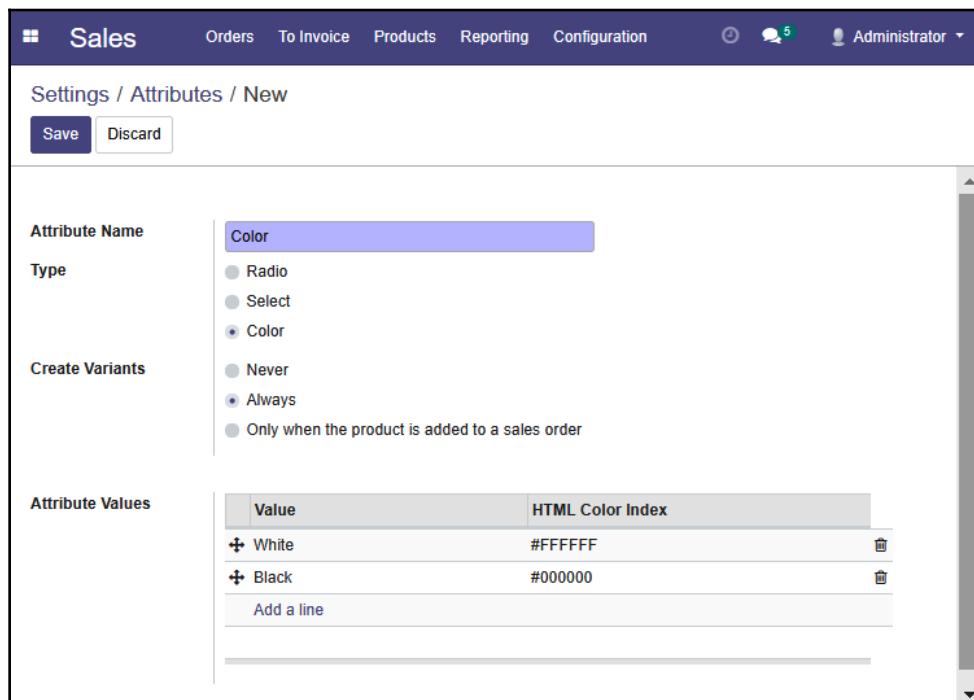
Creating routings based on product variants

Odoo allows you to create products that have variants. For example, we can have T-shirts that come in variants of color and size. In manufacturing operations, it can be common to require different processes based on the variant of the product. Screen printing is no different. You may need to alter the process if the shirt is white versus if the shirt is black. You may also have a different press for large T-shirts as opposed to small T-shirts. Let's see how you can configure Odoo to change the routing based on product variants.

First, you must go into the **Sales** application and, under **Settings**, check the **Variants and Options** option:



Now, you can click **Attributes** to set up some variants for us to use for our example. Click **Create** to make the first attribute. In Odoo 12, variants have been improved to allow more flexibility. Working with T-shirts, **Color** is a common variant. Fortunately, in Odoo 12, **Color** is a built-in variant type. Click **Create** and configure the color attributes, as follows:



Notice how, under **Type**, we have chosen **Color**. This will tell the Odoo framework to display the color that we specified when entering the order.

Controlling the creation of variants

A powerful new feature in Odoo 12 is to allow you to determine when you create your product variants. When setting up color and size, for example, you could have 12 different colors with 12 different sizes. This creates a great number of options that you may never purchase. If you choose **Always**, Odoo will create a product variant for every possible color and size combination. However, if you choose **Only when the product is added to a sales order**, Odoo will show you the available colors, but it will only actually create a variant when the item is added to a sales order.

Adding a size attribute

Now that we have added a **Color** attribute, let's add a variant for **Size**. After clicking **Create**, fill the form out, as follows:

The screenshot shows the Odoo interface for creating a new attribute. The top navigation bar includes 'Sales', 'Orders', 'To Invoice', 'Products', 'Reporting', 'Configuration', and 'Administrator'. Below the header, there are two buttons: 'Save' and 'Discard'. The main form has the following fields:

- Attribute Name:** Size
- Type:** Radio (selected)
- Create Variants:** Always (selected)
- Attribute Values:** A table with columns 'Value' and 'HTML Color Index'. It contains four rows: 'Small', 'Medium', 'Large', and 'Extra Large'. Each row has a delete icon on the right.

The **Radio** type allows the user to only check one option. If you use the **Select** type, then the user will be able to select multiple options:

The screenshot shows the Odoo Sales module's 'Attributes' configuration screen. At the top, there are tabs for 'Orders', 'To Invoice', 'Products', 'Reporting', and 'Configuration'. On the right, there are icons for a clock, a speech bubble with '5', and an administrator profile. Below the tabs, there is a search bar with placeholder text 'Search...' and a magnifying glass icon. To the right of the search bar are buttons for 'Filters', 'Group By', 'Favorites', and page navigation (1-2 / 2). There are also 'Create' and 'Import' buttons. The main area displays a table with columns for 'Attribute', 'Type', and 'Create Variants'. Two attributes are listed: 'Color' (Type: Color) and 'Size' (Type: Radio). The 'Size' row has a note 'Always' under 'Create Variants'.

Now that we have set up our **Attributes**, we can go to our product and specify the variant values in which our product is available. In this example, we have specified that the shirt can come in **Black or White**:

The screenshot shows the Odoo Products module's 'Class of 2017 T-Shirt' configuration screen. At the top, there are tabs for 'Orders', 'To Invoice', 'Products', 'Reporting', and 'Configuration'. On the right, there are icons for a clock, a speech bubble with '5', and an administrator profile. Below the tabs, there are buttons for 'Edit', 'Create', 'Print', and 'Action'. There are also buttons for 'Configure Variants', 'Update Qty On Hand', and 'Replenish'. The main area displays a summary for 'Class of 2017 T-Shirt' with sections for 'General Information' and 'Variants'. Under 'General Information', there are checkboxes for 'Can be Sold' (checked) and 'Can be Purchased' (unchecked). Under 'Variants', there is a summary table with columns for 'Variants' (2), 'Unit(s) On Hand' (0.000), 'Unit(s) Forecasted' (0.000), 'Product Moves' (0), 'Reordering R...' (0.00 Unit(s)), and 'Unit(s) Sold' (0.000). A 'More' button is also present. Below this, there are tabs for 'General Information', 'Variants' (which is selected), 'Sales', and 'Inventory'. The 'Variants' tab shows a table with columns for 'Attribute' and 'Attribute Values'. One row shows 'Color' with 'White' and 'Black' as attribute values. A warning message at the bottom states: 'Warning: adding or deleting attributes will delete and recreate existing variants and lead to the loss of their possible customizations.'

Now, we can configure a **Bill of Materials** and **Routing** based on which color we need to produce:

The screenshot shows the Odoo Sales module interface. At the top, there's a navigation bar with tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the right, there are icons for a clock, notifications (5), and a user named Administrator. Below the navigation bar, the page title is "Products / Class of 2017 T-Shirt / Bill of Materials / New". There are two buttons at the top left: "Save" and "Discard".

The main form has several input fields:

- Product:** Class of 2017 T-Shirt (dropdown)
- Product Variant:** Class of 2017 T-Shirt (Black) (dropdown)
- Quantity:** 1.00 (input field)
- Routing:** Print Job - Black (dropdown)

On the right side, there are two buttons: "Active" (green) and "Structure & Cost". Below these buttons is a section with two radio buttons:

- Manufacture this product
- Kit

At the bottom left, there are two tabs: "Components" (selected) and "Miscellaneous". The "Components" tab displays a table with one row:

Component	Quantity	Product Unit of Measure	Consumed in Operation
+ Medium T-Shirt	1.000	Unit(s)	

Below the table, there's a link "Add a line".

Notice how we were able to specify an alternate **Routing** for the **Black** product variant. Using this method, you can get a great deal of control over how you manage the production of your products.

Using unbuild orders or a reverse BoM

Sometimes, you will have a situation in which you purchase products that must be broken down and taken into **Inventory** as their constituent parts. One example may be a returned product that you then wish to disassemble back into its raw parts. Note that you must have an existing completed manufacturing order to create an **Unbuild Order**.

Here, we have created an **Unbuild Order** for a **Class of 2017 T-Shirt (Black)** manufacturing order:

The screenshot shows the 'Manufacturing' module interface. At the top, there are tabs for Overview, Operations, Master Data, Reporting, Configuration, and a notifications icon with 5 messages. On the right, there's an 'Administrator' dropdown. Below the tabs, it says 'Unbuild Orders / New'. There are 'Save' and 'Discard' buttons. A large button labeled 'Unbuild' is prominent. To its right are 'Draft' and 'Done' buttons. The main area is titled 'New' and contains fields for Product (set to 'Class of 2017 T-Shirt (Black)'), Manufacturing Order (set to 'WH/MO/00002'), Bill of Material (set to 'Class of 2017 T-Shirt'), and Quantity (set to '1.00 Unit(s)').

Once you click the **Unbuild** button to confirm the order, the white medium T-shirt that was used in the production of the original manufacturing order will be returned to **Inventory**.

Creating a kit using a BoM

Up until now, our BoM has been used to manufacture products. This process consumes the products in the BoM to produce an all-new product. There is, however, an entirely different way to use a BoM to create product kits. For example, let's say we have a **Fan Appreciation Bundle** that includes a **Hat**, **Medium White T-Shirt**, and a **Bumper Sticker**. When the customer purchases the **Fan Appreciation Bundle**, we don't want to create a new product; instead, we want to ship the individual products to the customer.

Here, we have created a **Bill of Materials** that has a kit with our example:

The screenshot shows the Odoo Manufacturing application interface. At the top, there's a navigation bar with tabs: Overview, Operations, Master Data, Reporting, Configuration, and a user icon for Administrator. Below the navigation bar, the main title is "Bills of Materials / New". There are two buttons: "Save" and "Discard". On the right side, there are two filter buttons: "Active" and "Structure & Cost". The main form contains the following fields:

- Product:** Fan Appreciation Bundle
- Product Variant:** (dropdown)
- Quantity:** 1.00 Unit(s)
- Routing:** (dropdown)
- BoM Type:** Reference (radio button)
- Manufacture this product:** (radio button)
- Kit:** (radio button) (selected)

A detailed explanation for the Kit type is provided:

- If the product is a finished product: When processing a sales order for this product, the delivery order will contain the raw materials, instead of the finished product.
- If the product is a semi-finished product: When processing a manufacturing order that contains that product as component, the raw materials of that product will be added to the manufacturing order of the final product.

Below the form, there are two tabs: "Components" (selected) and "Miscellaneous". The "Components" tab displays a table with three rows:

Component	Quantity	Product Unit of Measure	Apply on Variants	Consumed in Operation
+ Hat	1.000	Unit(s)		
+ Medium White T-Shirt	1.000	Unit(s)		
+ Bumper Sticker	1.000	Unit(s)		

Add a line

Notice that, under **BoM Type**, we have chosen **Kit**. We can also see that Odoo provides a detailed explanation as to how it will handle the kits. Since our example is a simple kit, the delivery order will simply contain the **Components** we have specified.

Summary

In this chapter, we installed the Manufacturing application so that we could set up our manufacturing process. A BoM was created to define what products would be consumed when our product was manufactured. We manufactured our final product and looked at the inventory analysis report to verify our results. We then turned on the work center feature so that we could explore how to set up work centers and routings to get more control over our manufacturing operations.

In the next chapter, we will take a closer look at accounting and other reporting options. Setting up your chart of accounts is an important step that we'll cover, as well as reviewing journal entries, creating invoices, and receiving payments. We will also be defining sales taxes and managing fiscal periods. Yes, there is a lot more to cover!

6

Configuring Accounting - Finance Options

One of the nice things about Odoo is that you can get up and running quickly without having to spend a lot of time setting up complicated accounting and finance options. Odoo does a reasonable job of creating a basic chart of account structures as a starting point, helping you to get familiar with Odoo. When setting up a production system for your company, however, you will want to take the time to properly define your accounting requirements.

In this chapter, we will learn about how to configure accounting in Odoo by covering the following topics:

- Examining and defining the chart of accounts
- Enabling the user to see all accounting features
- Stepping through the complete sales process
- Creating new accounts, viewing reports, and making journal entries

Defining the chart of accounts for your business

The backbone of an accounting system setup is the chart of accounts. Wikipedia defines a chart of accounts as follows:

"A chart of accounts is a created list of the accounts that are used by a business entity to define each class of items for which money or the equivalent is spent or received. It is used to organize the finances of the entity and to segregate expenditures, revenue, assets, and liabilities in order to give interested parties a better understanding of the financial health of the entity."

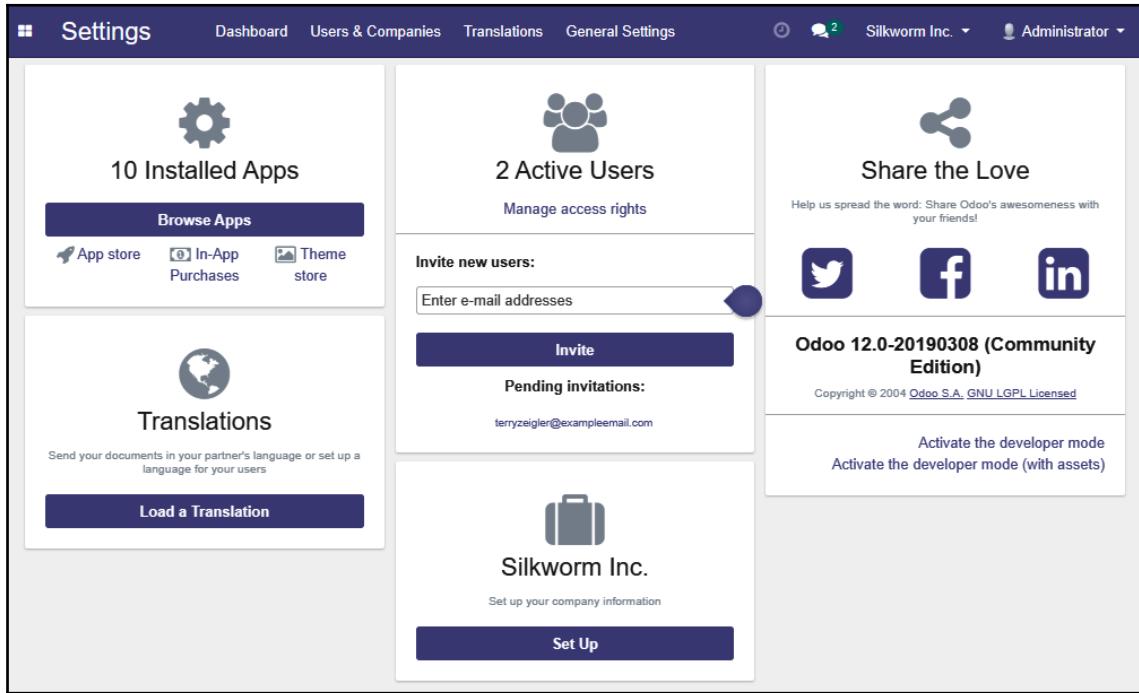
It is very likely that if you are setting up Odoo for an existing business, you will be asked to configure the chart of accounts in Odoo to match the account structure the business is already using. Even if you are not tied to any existing chart of accounts, it is inevitable that you will need to have a firm understanding of how the accounting functionality in Odoo works if you are going to have a successful implementation.

If you are completely unfamiliar with accounting, then this chapter may prove somewhat challenging. It is important to get familiar with the basics of accounting if you want to succeed in implementing any ERP system. In the [Appendix A, Locating Additional Odoo Resources](#), you can find resources that can help you get started with learning basic accounting.

Configuring a user to see all Community Edition accounting options

Odoo configures a basic accounting structure when you install base modules such as **Sales and Purchasing**. Prior to Odoo 11, to access all of the accounting configuration options, you needed to install the **Accounting & Finance** application. In Odoo 12, you have access to a limited set of accounting and finance options, and the reports are delivered in PDF files, making it a little bit more awkward to work with. Still, you can unlock the most important options that are tucked away inside of the Odoo developer mode.

We will begin by navigating to the **Settings** menu and then clicking **Activate the developer mode** on the far right of the dashboard:



In this case, we are turning on developer mode just so that we can see all the options that are available for a user. Without developer mode turned on, we won't see the necessary options for a given user.

Next, you need to click Users under the **Users & Companies** menu and select the user for whom you wish to allow access to the full accounting features. In this case, we have selected the **Administrator**:

The screenshot shows the Odoo application interface for managing users. At the top, there's a navigation bar with links for Settings, Dashboard, Users & Companies, Translations, General Settings, Technical, and an administrator icon. Below the navigation is a toolbar with buttons for Edit, Create, Action, and pagination (1/1). A status bar indicates 'Never Connected' and 'Confirmed'. The main content area is titled 'Users / Administrator' and shows a single user record for 'Administrator'. The user has a placeholder profile picture, the name 'Administrator', and the login 'admin'. The status is marked as 'Active'. Below the basic information, there are tabs for 'Access Rights' and 'Preferences', with 'Access Rights' currently selected. Under 'User Type', it shows 'User types' as 'Internal User'. The 'Application Accesses' section lists various modules and their assigned roles: Sales (Manager), Project (Manager), Inventory (Manager), Manufacturing (Manager), Accounting & Finance (Billing Manager), Purchases (Manager), Employees (Manager), Leaves (Manager), Timesheets (Manager), Website (Editor and Designer), and Administration (Settings). The 'Technical Settings' section contains several checkboxes for various accounting features, with some being checked (e.g., 'Allow the cash rounding management', 'Analytic Accounting', 'Discount on lines').

Scroll down and you will find the **Show Full Accounting Features** option. Check this option to enable the full accounting options in Odoo 12:

The screenshot shows the Odoo Settings page under the 'Users / Administrator' tab. The page lists various configuration options in three sections: 'Analytic Accounting Tags', 'Extra Rights', and 'Other'. In the 'Other' section, there is a checkbox labeled 'Show Full Accounting Features'. This checkbox is checked, indicating it has been enabled.

Section	Option	Status
Analytic Accounting Tags	Discount on lines	<input type="checkbox"/>
	Display incoterms on Sales Order and related invoices	<input type="checkbox"/>
	Manage Different Stock Owners	<input type="checkbox"/>
	Manage Multiple Stock Locations	<input type="checkbox"/>
	Manage Multiple Warehouses	<input type="checkbox"/>
	Manage Pricelist Items	<input type="checkbox"/>
	Manage Product Variants	<input checked="" type="checkbox"/>
	Manage Vendor Price	<input type="checkbox"/>
	Manage delivery dates from sales orders.	<input type="checkbox"/>
	Pro-forma Invoices	<input type="checkbox"/>
Extra Rights	Show Full Accounting Features	<input checked="" type="checkbox"/>
	Use products on vendor bills	<input checked="" type="checkbox"/>
	Multi Companies	<input type="checkbox"/>
	Quotation Templates	<input type="checkbox"/>
Other	Tax display B2C	<input type="checkbox"/>
	Tax display B2B	<input checked="" type="checkbox"/>

After you have configured the user to access the full accounting features, the contents of the **Invoicing** menu will change to include far more choices. Like when you're changing other settings, it's possible that you may need to use *Shift-refresh* in your browser to see the changes. Unlike previous versions of Odoo, the menu will remain as **Invoicing**, even when all of the options are enabled.

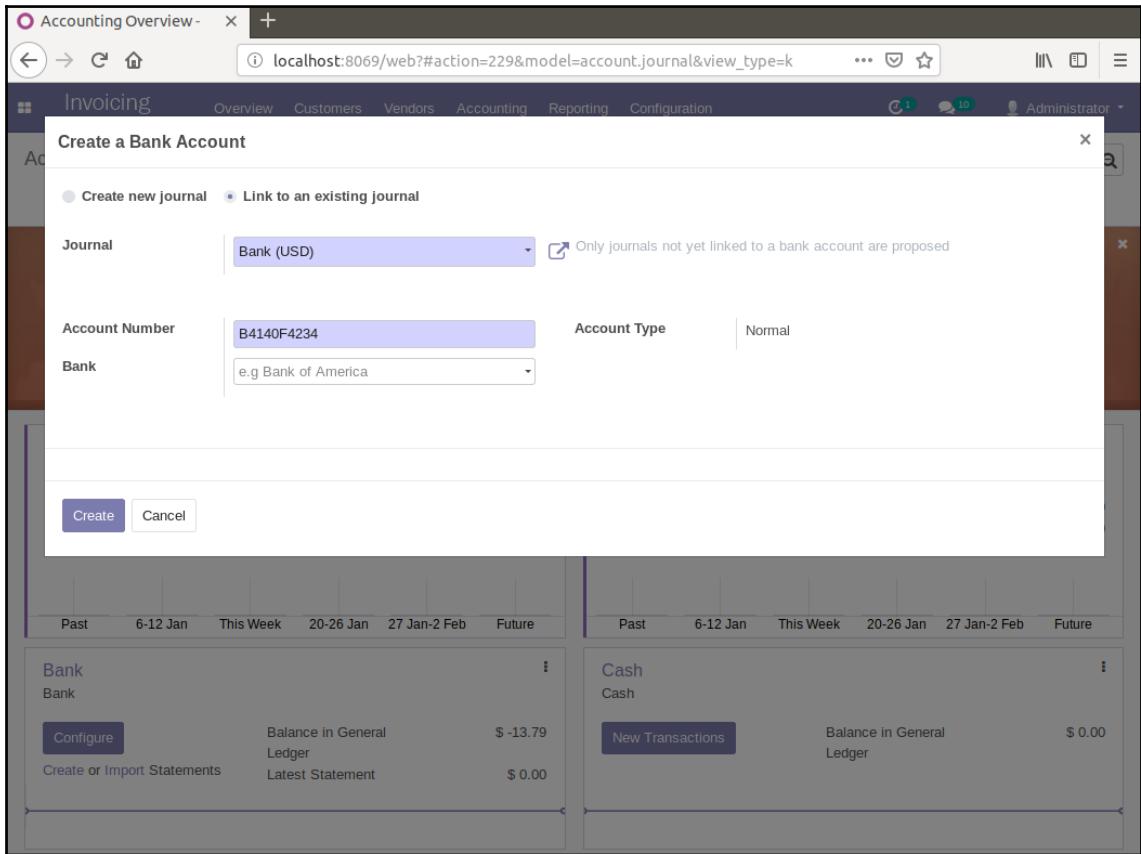
Using the configuration panel for accounting

Even though the option for full accounting features is buried deep in the Odoo user access rights, you still get full access to the **Accounting & Finance** configuration panel that was introduced in Odoo 12. Please note that you may have to hit *Shift-refresh* in your browser to see the new accounting menus and options:

The screenshot shows the 'Accounting Overview' configuration panel in Odoo. At the top, there's a navigation bar with tabs for Invoicing, Overview, Customers, Vendors, Accounting, Reporting, and Configuration. The Configuration tab is active. Below the navigation is a search bar and filter options. The main area is titled 'Accounting Overview' and contains four sections: 'Company Data', 'Bank Account', 'Fiscal Year', and 'Chart of Accounts'. Each section has a brief description and a status indicator (e.g., 'All done!'). Below these are two columns of cards: 'Customer Invoices' and 'Vendor Bills' under 'Purchase', and 'Bank' and 'Cash' under 'Bank'. Each card provides summary information and action buttons like 'New Invoice', 'Import Bills', 'Configure', and 'New Transactions'.

You will see that **Company Data** is already checked off as we completed that earlier when we installed the **Sales** application. To proceed with configuring accounting, it is best that we finish the remaining steps.

Click **Add a Bank** to add a bank account to our Odoo implementation:



You are provided with two options. You can create a new journal, or you can link the bank account to an existing journal. Since we have already created transactions, we have configured the account to link to the existing bank journal.

Next, you can tie the account to the actual bank. This basically creates a contact record with an address for the bank.

Click **Create** to create the bank account:

The screenshot shows the 'Accounting Overview' page in a web application. At the top, there are tabs for Invoicing, Overview, Customers, Vendors, Accounting, Reporting, Configuration, and Administrator. The 'Invoicing' tab is selected. The main area is titled 'Accounting Overview' and features a progress bar with four steps: 'Company Data' (All done!), 'Bank Account' (Step Completed!), 'Fiscal Year', and 'Chart of Accounts'. Below the progress bar, there are sections for 'Customer Invoices' (Sale), 'Vendor Bills' (Purchase), 'Bank', and 'Cash'. Each section includes buttons for 'New Invoice', 'Import Bills', 'Configure', and 'Create or Import Statements'. The 'Bank' section shows a balance of \$ -13.79. The 'Cash' section shows a balance of \$ 0.00. The bottom of the screen has navigation buttons for '1-4 / 4'.

Note that the screen will refresh, and you can confirm that you have completed this step.

Next, we will configure the **Fiscal Year**:

The screenshot shows the Odoo Accounting Overview interface. A modal window titled 'Fiscal Year' is open, allowing the user to set the 'Opening Date' to 01/01/2018 and the 'Fiscal Year End' to December 31. Below the modal, there are status messages: 'All done!' with a checkmark and 'Step Completed!' with a checkmark. There are also 'Configure' and 'Review' buttons. The main dashboard displays four sections: Customer Invoices (Sale), Vendor Bills (Purchase), Bank, and Cash. Each section provides summary information and buttons for 'New Invoice', 'Import Bills', 'Configure', and 'Create or Import Statements'.

Here, we have specified the opening date of **01/01/2018** and our **Fiscal Year End** as **December 31**. Odoo does not require any period closings or end-of-year closings. You simply post your transactions to the date they have occurred and Odoo will handle the rest.



In my experience, many accountants will need some Odoo consultation and training regarding how Odoo manages the chart of accounts, period closings, and year-end procedures. Many are used for alternative procedures and processes regarding how the month end and year end are managed.

Reviewing the current chart of accounts

Now, we will view the **Chart of Accounts** in Odoo.

In addition to accessing the **Chart of Accounts** through the configuration wizard, you can also go to the **Invoicing** menu and choose **Chart of Accounts** under the **Configuration | Accounting** submenu. You will immediately see the chart of accounts sorted by the **Code** column:

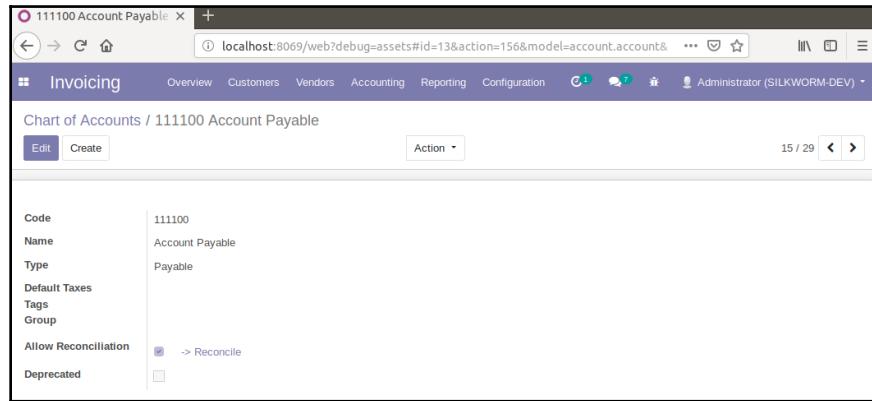
<input type="checkbox"/>	Code	Name	Type
<input type="checkbox"/>	100000	Fixed Asset Account	Fixed Assets
<input type="checkbox"/>	101000	Current Assets	Current Assets
<input type="checkbox"/>	101110	Stock Valuation Account	Current Assets
<input type="checkbox"/>	101120	Stock Interim Account (Received)	Current Assets
<input type="checkbox"/>	101130	Stock Interim Account (Delivered)	Current Assets
<input type="checkbox"/>	101200	Account Receivable	Receivable
<input type="checkbox"/>	101300	Tax Paid	Current Assets
<input type="checkbox"/>	101401	Bank	Bank and Cash
<input type="checkbox"/>	101501	Cash	Bank and Cash
<input type="checkbox"/>	101600	Opening Income Account	Other Income
<input type="checkbox"/>	101701	Liquidity Transfer	Current Assets
<input type="checkbox"/>	102000	Non-current assets	Non-current Assets
<input type="checkbox"/>	103000	Prepayments	Prepayments
<input type="checkbox"/>	111000	Current Liabilities	Current Liabilities
<input type="checkbox"/>	111100	Account Payable	Payable
<input type="checkbox"/>	111200	Tax Received	Current Liabilities
<input type="checkbox"/>	111300	Reserve and Profit/Loss Account	Current Liabilities
<input type="checkbox"/>	112000	Non-current Liabilities	Non-current Liabilities
<input type="checkbox"/>	200000	Product Sales	Income

In the preceding screenshot, you can see the currently configured **Chart of Accounts**, including the **Code**, **Name**, and **Type** of account.

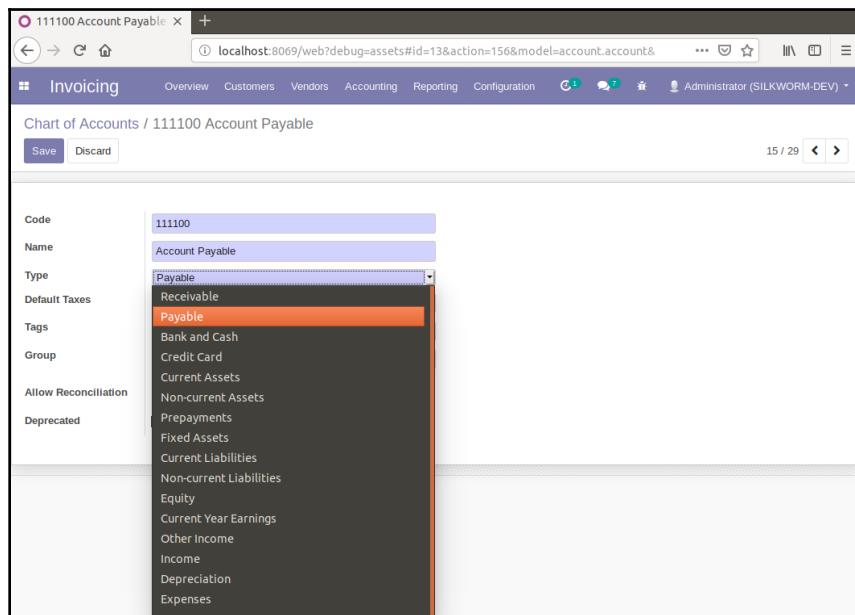
Getting more information on a specific account

In Odoo 12, the type of account is used by the various reports to determine whether that account should be included, and if so, where on the report to include that account. Clicking on one of the accounts will bring up the account, along with an example report structure so that you can see just how that account type will be represented.

Click on the **Account Payable** account in the **Chart of Accounts** list view:



Here, we can define an account and view information on a specific account. A very important field in this form to take note of is the **Type** field for the account. This field specifies how the account you create will get treated in transactions and various reporting features in Odoo:



The preceding screenshot shows that we have selected the **Payable** option for the **Type** field.

Completing the accounting configuration

Let's have a look at the following screenshot:

The screenshot shows the Odoo Accounting Overview page. At the top, there is a navigation bar with tabs for Invoicing, Overview, Customers, Vendors, Accounting, Reporting, and Configuration. On the right side of the header, there are icons for favorites, search, notifications (1), messages (10), and administrator settings.

The main content area is titled "Accounting Overview". It features a progress bar with four steps: "Company Data" (Setup your company's data for reports headers), "Bank Account" (Setup your bank account to sync bank feeds), "Fiscal Year" (Define your fiscal years opening & closing dates), and "Chart of Accounts" (Setup your chart of accounts and record initial balances). Each step has a checkmark indicating it is completed.

A message box in the center says "Nice work! Your configuration is done." with a "Close" button.

The bottom section contains four cards:

- Customer Invoices**: Sale. New Invoice button. 0 Invoices to Validate, 0 Unpaid Invoices. \$ 0.00.
- Vendor Bills**: Purchase. New Bill button. Import Bills button. 0 Bills to Validate, 0 Bills to Pay. \$ 0.00.
- Bank**: Bank. Configure button. Create or Import Statements button. Balance in General Ledger: \$ -13.79. Latest Statement: \$ 0.00.
- Cash**: Cash. New Transactions button. Balance in General Ledger: \$ 0.00.

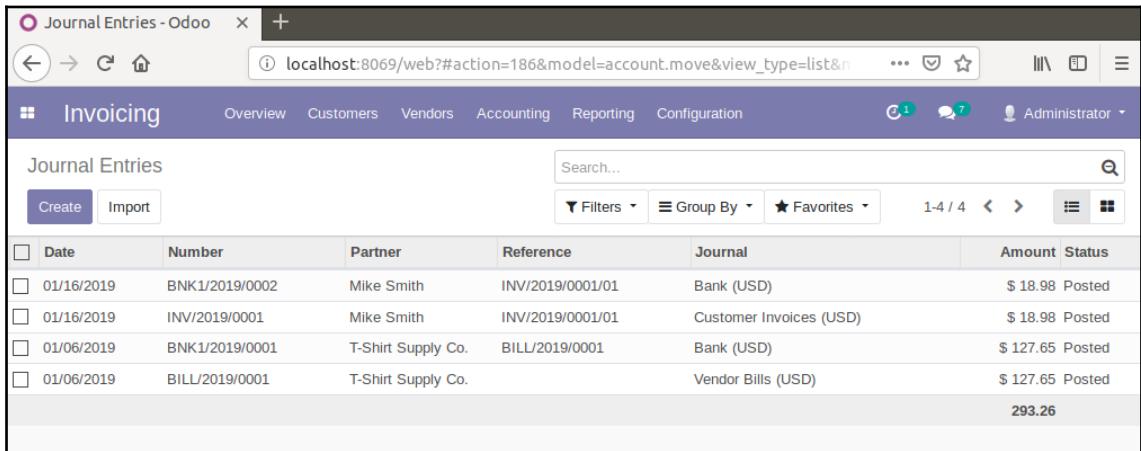
Below each card is a timeline with categories: Past, 6-12 Jan, This Week, 20-26 Jan, 27 Jan-2 Feb, Future.

Once you've finished looking through the accounts and returned to the main **Accounting Overview**, you will see that the configuration wizard has completed.

Learning how transactions in Odoo get posted to accounts

All transactions that take place in Odoo create **Journal Entries** that either credit or debit a specific account. Each journal entry must balance. This means that the debits must equal the credits. Odoo makes it very easy to examine your **Journal Entries** so that you can see exactly where each transaction is posted.

To view the **Journal Entries**, go to the **Invoicing** menu, and under the **Adviser** submenu, click **Journal Entries**. By default, Odoo applies a filter to restrict the journal entries to only **Miscellaneous Operations**. Clear this filter to see all the journal entries:



The screenshot shows the Odoo web interface for 'Journal Entries'. The top navigation bar includes links for Overview, Customers, Vendors, Accounting, Reporting, and Configuration, along with user status and administrator dropdowns. The main content area is titled 'Journal Entries' with buttons for 'Create' and 'Import'. A search bar and various filtering options are available. The data table lists four journal entries:

Date	Number	Partner	Reference	Journal	Amount	Status	
01/16/2019	BNK1/2019/0002	Mike Smith	INV/2019/0001/01	Bank (USD)	\$ 18.98	Posted	
01/16/2019	INV/2019/0001	Mike Smith	INV/2019/0001/01	Customer Invoices (USD)	\$ 18.98	Posted	
01/06/2019	BNK1/2019/0001	T-Shirt Supply Co.	BILL/2019/0001	Bank (USD)	\$ 127.65	Posted	
01/06/2019	BILL/2019/0001	T-Shirt Supply Co.		Vendor Bills (USD)	\$ 127.65	Posted	
						293.26	

In the preceding screenshot, you can see four journal entries.



If you don't see any journal entries in your list, make sure that you have cleared the **Miscellaneous Operations** filter; Odoo applies it by default.

You can see that, in the journal listing, you get a summary, including the amount and the status indicating whether the journal entry is posted or unposted. You can also see that, in the invoice journal entry, we have a reference that's provided for the original sales order.

Note that the journal entry summary doesn't contain any references to specific accounts in the **Chart of Accounts**. To see these details, we must click on a specific journal entry. Click on the invoice for \$18.98 that's assigned to our customer/partner Mike Smith.

You will now see the details of the transaction and each account that was involved:

The screenshot shows the Odoo Invoicing module. At the top, there's a header bar with tabs for Overview, Customers, Vendors, Accounting, Reporting, Configuration, and a user dropdown for Administrator. Below the header is a toolbar with buttons for Edit, Create, Reverse Entry, Action, and navigation arrows. A status bar at the bottom indicates '2 / 4' and 'Unposted'.

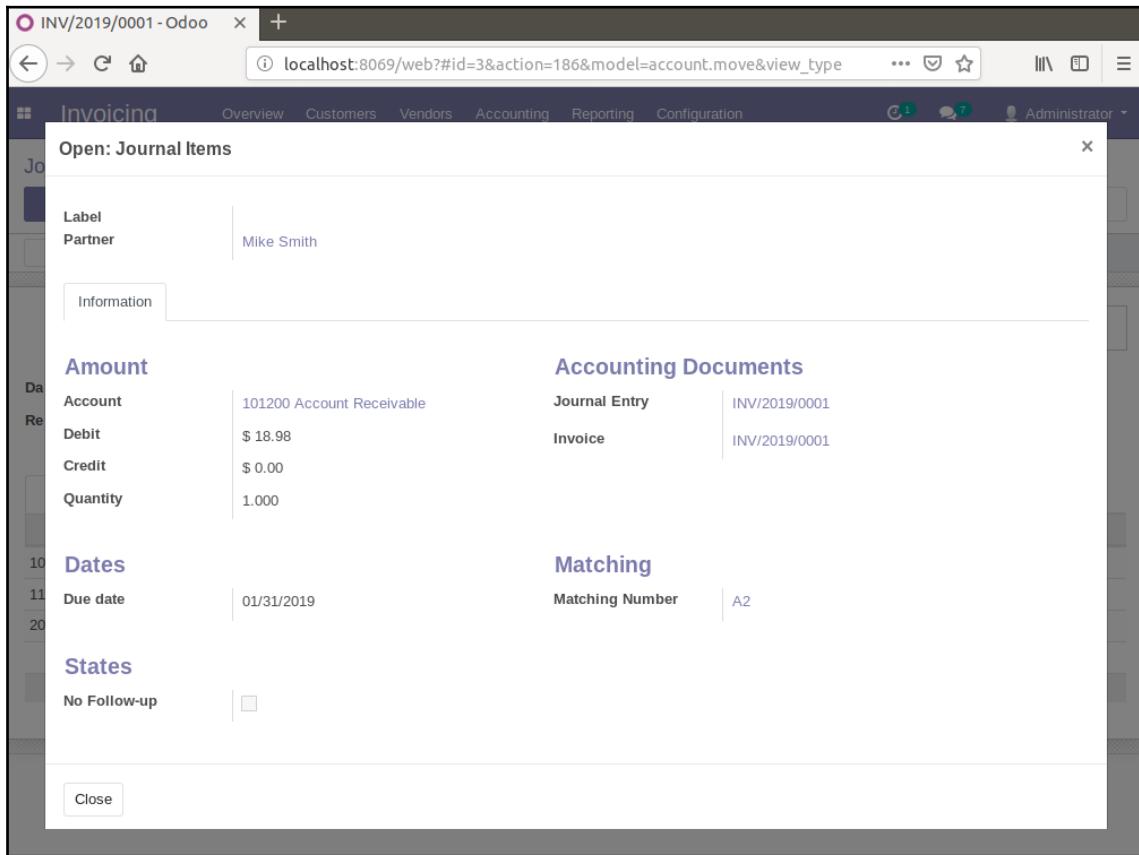
The main content area displays a journal entry titled 'INV/2019/0001'. It includes fields for Date (01/16/2019), Reference (INV/2019/0001/01), Journal (Customer Invoices (USD)), and a 'Reconciled entries' button.

The 'Journal Items' tab is selected, showing a table of transaction details:

Account	Partner	Label	Debit	Credit	Taxes Applied
101200 Account Receivable	Mike Smith		\$ 18.98	\$ 0.00	
111200 Tax Received	Mike Smith	Tax 15.00%	\$ 0.00	\$ 2.48	
200000 Product Sales	Mike Smith	Medium White T-Shirt	\$ 0.00	\$ 16.50	Tax 15.00%
			18.98	18.98	

When we open the account, we see all the postings to the specific accounts that were involved. In the first row, we can see that **101200 Accounts Receivable** was debited \$18.98 for the invoiced T-shirt order. Next, in the second row, Odoo has posted a credit of \$2.48 sales tax on the order. This goes into **111200 Tax Received**. Finally, in the third row, we can see that a \$16.50 credit was posted to **200000 Product Sales** for the T-shirts themselves. Note how the **Debit** column matches the **Credit** column. Debits and credits must always be equal in a journal entry.

More details of the transaction are displayed under each item. Click on the first posting to **101200 Account Receivable** to pull up more details on that posting:



In the preceding screenshot, you can use the links to quickly find the partner, account, journal, and journal entry related to the posting.

If this is still a little confusing, don't worry. We are now going to go through a set of transactions from the **Accounts Receivable** side so that you can better understand how Odoo handles accounting transactions.

Following transactions through the sales and accounts receivable process

In the previous example, we were looking at the chart of accounts and determining what transactions created the entries. Next, we will sell an item to a customer and see exactly how that transaction affects the accounting entries in the journal.

Let's begin by creating a new sales order.

Go to **Sales** and click on **Orders** to bring up the sales order listing. Click on **Create** to create a new sales order:

The screenshot shows the Odoo Sales Orders interface for SO003. At the top, there are buttons for Edit, Create, Print, and Action. Below that is a toolbar with Create Invoice, Preview, Print, Send by Email, Cancel, Lock, Quotation, Quotation Sent, and Sales Order. The main area displays the details of SO003, including the customer (Mike Smith, 444 South Main, Marion IL 62959), confirmation date (01/16/2019 15:41:48), and payment terms (15 Days). A delivery icon indicates 1 delivery. The Order Lines section shows one item: Medium White T-Shirt, Ordered Qty: 5.00, Delivered Quantity: 0.000, Invoiced Quantity: 0.000, Unit of Measure: Unit(s), Unit Price: 16.50 (Tax 15.00%), and Subtotal: \$ 82.50. At the bottom, financial summary lines show Untaxed Amount: \$ 82.50, Taxes: \$ 12.38, and Total: \$ 94.88.

If you have followed along with our examples, then you will already have the customer and product entered to create the sample sales order. Otherwise, you will need to add a customer and a product, if you wish to follow along on your computer. In this example, we have created a sales order for five **Medium White T-Shirts**. Make sure that you click **Confirm Sale** to create the sales order.



Odoo will automatically number sales orders and other documents. In the preceding example, two sales order numbers have already been used by the Odoo system. Therefore, depending on what you have already done with your current system, you may not have the same sales order number for your sample.

At this point, if you were to go and look at the journal entry listing, you would not see any additional journal entries. *Why is this?* The way Odoo is currently configured means that we must manually create an invoice. As long as you are in the **Sales Order** status, you will not see any transactions in accounting.

Only when we click on the **Create Invoice** button at the top of the screen will Odoo start the process to create accounting transactions.

Click on **Create Invoice** to generate a draft invoice for this sales order:

Product	Description	Ordered Qty	Delivered Quantity	Invoiced Quantity	Unit of Measure	Unit Price	Taxes	Subtotal
Medium White T-Shirt	Medium White T-Shirt	5.000	0.000	0.000	Unit(s)	16.50	Tax 15.00%	\$ 82.50

Odoo will present you with a wizard that allows you to determine how you wish to invoice. Please note that the primary difference is in how down payments are handled. Since we have no down payments, we are fine to take the default option of **Invoiceable lines** (deduct down payments). If we did have a down payment, then it would be automatically deducted from the invoice total.

Click on **Create and View Invoices**:

The screenshot shows the Odoo Sales module interface. At the top, there's a header bar with tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the right side of the header, there are icons for a user (Administrator), notifications (9), and other system functions. Below the header, the main content area is titled "Sales Orders / SO003 / Invoice". It has buttons for Edit, Create, Print, Action, Validate, Preview, Draft, Open, and Paid. The "Draft Invoice" section contains fields for Customer (Mike Smith, 444 South Main, Marion IL 62959), Payment Terms (15 Days), Invoice Date, Due Date, Salesperson (Administrator), and Sales Team (Sales). The "Invoice Lines" tab is selected, showing a single line item for a Medium White T-Shirt. The line item details are: Product (Medium White T-Shirt), Description (Medium White T-Shirt), Account (200000 Product Sales), Quantity (5.000 Unit(s)), Price (16.50), Taxes (Tax 15.00%), and Subtotal (\$ 82.50). At the bottom of the invoice lines, there are summary totals: Untaxed Amount (\$ 82.50), Tax (\$ 12.38), and Total (\$ 94.88). The footer of the page includes links for Send message, Log note, Schedule activity, and Today, along with a Following button and a user count of 2.

Because this is just a **Draft Invoice**, if you look at the journal entries, you will see no changes. However, if you look at the **Draft Invoice**, you will see the transactions that will be created once you validate the invoice.

In the line item of the invoice, you will see the **200000 Product Sales** account. This will be the account that will be credited for the sale of the medium white T-shirts the customer has purchased.

Choose the **Other Info** page on the **Invoice** form. Note that under **Account**, it reads **101200 Account Receivable**; this account will be debited to record the amount the customer owes to the company once the invoice is generated.

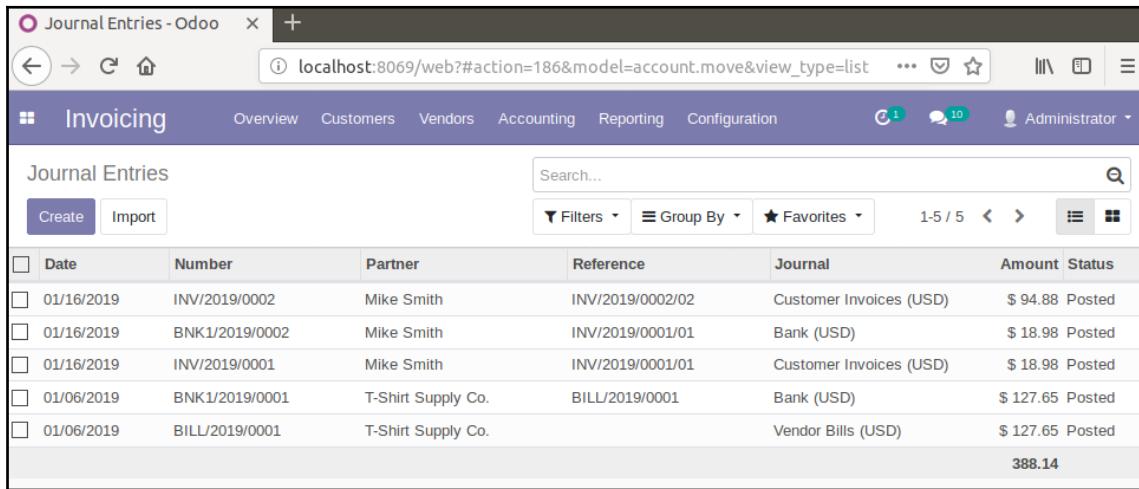
You can also see that this invoice will post **\$12.38** to the **111200 Tax Received** account:

The screenshot shows the Odoo interface for a draft invoice. At the top, there's a header bar with tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the right, there are notifications for 1 message and 9 comments, and a dropdown for Administrator. Below the header, the page title is "Sales Orders / SO003 / Invoice". There are buttons for Edit, Create, Print, Action, Validate (which is highlighted), and Preview. A status bar indicates the invoice is Draft, with Open and Paid options available. The main content area is titled "Draft Invoice". It contains two sections of data tables. The first section includes fields for Customer (Mike Smith, 444 South Main, Marion IL 62959), Payment Terms (15 Days), and Sales Team (Sales). The second section includes fields for Journal (Customer Invoices (USD)), Account (101200 Account Receivable), Payment Ref. (Reference/Description), Incoterm, Fiscal Position, Source Document (SO003), and Bank Account. At the bottom, a table shows a tax line item: Tax Description (Tax 15.00%), Tax Account (111200 Tax Received), and Amount Total (\$ 12.38). At the very bottom, there are buttons for Send message, Log note, Schedule activity, and a Following indicator with 2 users. The date Today is shown, and a footer note says "Administrator - a minute ago".

Click on **Validate** to post the invoice and create the transactions.

Viewing the transactions that have been created by validating the invoice

Now that we have validated our invoice, Odoo has automatically created the accounting transactions to increase our **Accounts Receivable** assets and the accounting transaction to record the sale. We can now open the journal entries back up and see the newly posted transaction:



The screenshot shows the Odoo web interface for 'Journal Entries'. The top navigation bar includes links for Invoicing, Overview, Customers, Vendors, Accounting, Reporting, Configuration, and Administrator. The main content area is titled 'Journal Entries' and displays a list of five entries. The columns are Date, Number, Partner, Reference, Journal, Amount, and Status. The entries are:

Date	Number	Partner	Reference	Journal	Amount	Status
01/16/2019	INV/2019/0002	Mike Smith	INV/2019/0002/02	Customer Invoices (USD)	\$ 94.88	Posted
01/16/2019	BNK1/2019/0002	Mike Smith	INV/2019/0001/01	Bank (USD)	\$ 18.98	Posted
01/16/2019	INV/2019/0001	Mike Smith	INV/2019/0001/01	Customer Invoices (USD)	\$ 18.98	Posted
01/06/2019	BNK1/2019/0001	T-Shirt Supply Co.	BILL/2019/0001	Bank (USD)	\$ 127.65	Posted
01/06/2019	BILL/2019/0001	T-Shirt Supply Co.		Vendor Bills (USD)	\$ 127.65	Posted
						388.14

The number assigned to this specific invoice is **INV/2019/0002** in the amount of **\$94.88**.

We can now click on this **Journal Entry** to see the details of the transaction:

The screenshot shows the Odoo Invoicing module's Journal Entries screen for INV/2019/0002. The top navigation bar includes links for Overview, Customers, Vendors, Accounting, Reporting, Configuration, and Administrator. The main header displays 'Journal Entries / INV/2019/0002'. Below the header are buttons for Edit, Create, Action, Reverse Entry, Unposted, and Posted. The journal entry details section shows the date (01/16/2019), reference (INV/2019/0002/02), journal (Customer Invoices (USD)), and a 'Reconciled entries' button. The 'Journal Items' tab is selected, displaying a table of transaction details:

Account	Partner	Label	Debit	Credit	Taxes Applied
101200 Account Receivable	Mike Smith		\$ 94.88	\$ 0.00	
111200 Tax Received	Mike Smith	Tax 15.00%	\$ 0.00	\$ 12.38	
200000 Product Sales	Mike Smith	Medium White T-Shirt	\$ 0.00	\$ 82.50	(Tax 15.00%)
			94.88	94.88	

In **Journal Items**, you can now see the same figures we examined on the invoice that was posted with the correct accounts. The **101200 Account Receivable** account has been debited by **\$94.88** to show the new current asset that represents this customer invoice. The customer owes the company **\$94.88**. As you create invoices and customers owe you money, **Accounts Receivable** will continue to grow.

Next, you will see that the tax for the invoice is a credit to the **111200 Tax Received** account. Typically, you will use this **Tax Received** account to later send that money on to the appropriate government agency. This, by the way, would require you to write a check that would credit your bank account (reducing its value) while posting a debit to the **Tax Received** account.

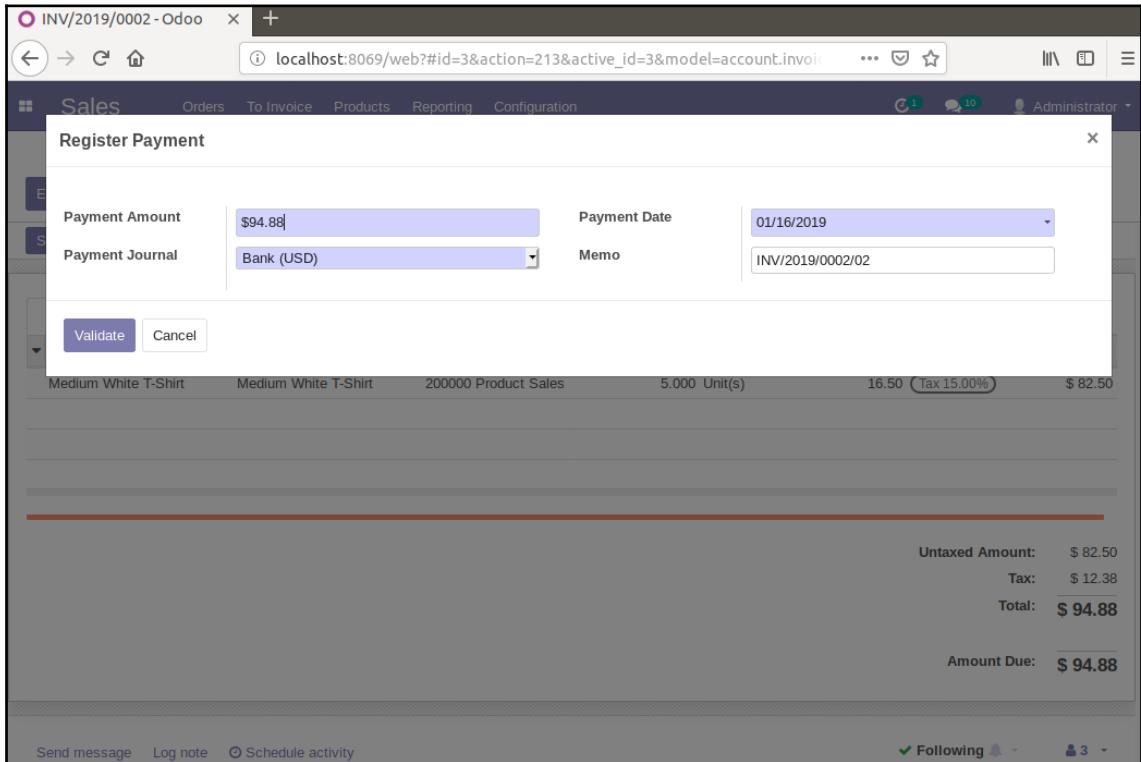
Finally, note that the **200000 Product Sales** account has been credited with **\$82.50**. This account will continue to be credited for products you sell.



For our example, we are using only one sales account to keep things simple. In most companies, you will have far more sales accounts to organize the various types of products that are sold.

Now, let's see what happens to these accounts when a customer pays their invoice.

Go to **Accounting** and choose **Customer Invoices**, then click on the invoice to bring up the form. Click on **Register Payment** to bring up the **Register Payment** form:

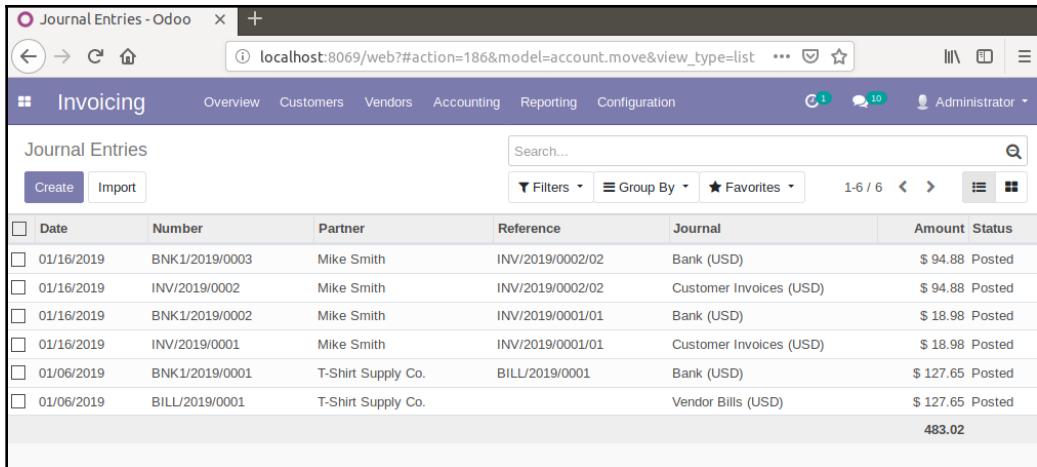


For our example, we have chosen the **Bank (USD)** payment method. You have the option to provide a memo to document that invoice payment. By default, this will include the sales order number, but many businesses may wish to include a check number as well.

Click the **Validate** button to pay the invoice and create the appropriate accounting transactions.

The invoice is now paid and the journal entries have been automatically created, and we can now see the payment listed as new **Journal Entries**.

Like you have done previously, use the **Accounting** menu and choose **Journal Entries** to bring up the list of **Journal Entries**:

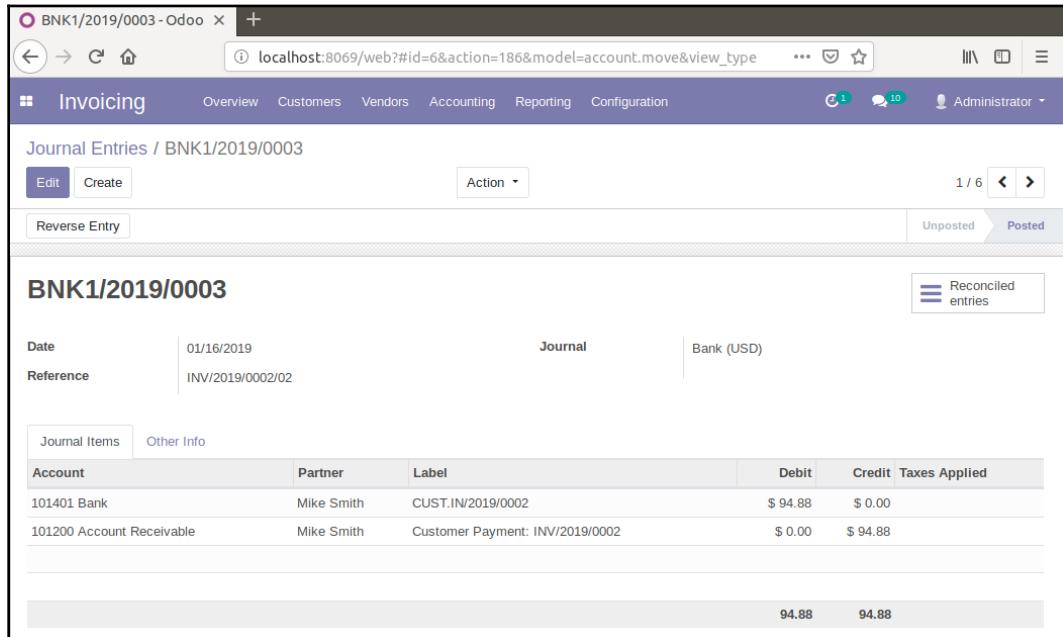


The screenshot shows the Odoo Journal Entries list view. The table has columns: Date, Number, Partner, Reference, Journal, Amount, and Status. One entry is highlighted: BNK1/2019/0003, dated 01/16/2019, for Mike Smith, reference INV/2019/0002/02, journal Bank (USD), amount \$94.88, and status Posted.

Date	Number	Partner	Reference	Journal	Amount	Status
01/16/2019	BNK1/2019/0003	Mike Smith	INV/2019/0002/02	Bank (USD)	\$ 94.88	Posted
01/16/2019	INV/2019/0002	Mike Smith	INV/2019/0002/02	Customer Invoices (USD)	\$ 94.88	Posted
01/16/2019	BNK1/2019/0002	Mike Smith	INV/2019/0001/01	Bank (USD)	\$ 18.98	Posted
01/16/2019	INV/2019/0001	Mike Smith	INV/2019/0001/01	Customer Invoices (USD)	\$ 18.98	Posted
01/06/2019	BNK1/2019/0001	T-Shirt Supply Co.	BILL/2019/0001	Bank (USD)	\$ 127.65	Posted
01/06/2019	BILL/2019/0001	T-Shirt Supply Co.		Vendor Bills (USD)	\$ 127.65	Posted
					483.02	

Note that we now have an entry number of **BNK1/2019/0003** that is also for **\$94.88**, the same amount as the invoice. The **Journal** allows you to easily organize transactions and identify exactly which accounts will be affected.

Let's examine the details of the payment that has been posted by clicking on the journal entry:



The screenshot shows the Odoo Journal Entry details view for BNK1/2019/0003. The table has columns: Date, Reference, Journal, and Bank (USD). The entry is dated 01/16/2019, reference INV/2019/0002/02, journal Bank (USD), and amount \$94.88.

Date	Journal	Bank (USD)
01/16/2019	INV/2019/0002/02	

Below the table, there are tabs for Journal Items and Other Info. The Journal Items tab shows two entries:

Account	Partner	Label	Debit	Credit	Taxes Applied
101401 Bank	Mike Smith	CUST.IN/2019/0002	\$ 94.88	\$ 0.00	
101200 Account Receivable	Mike Smith	Customer Payment: INV/2019/0002	\$ 0.00	\$ 94.88	

The total debit and credit amounts are both \$94.88.

In this cash receipt, you will notice that we can see the details as to exactly which accounts will be affected when we post the entry:

- **101401 Bank** is debited with **\$94.88**. This will increase this asset account.
- **101200 Account Receivable** is credited with **\$94.88**. This will decrease this asset account.

Essentially, this journal entry transfers the potential asset the customer owes the company from **Account Receivable** into the **Bank** account. The customer's account balance is reduced to reflect their payment.

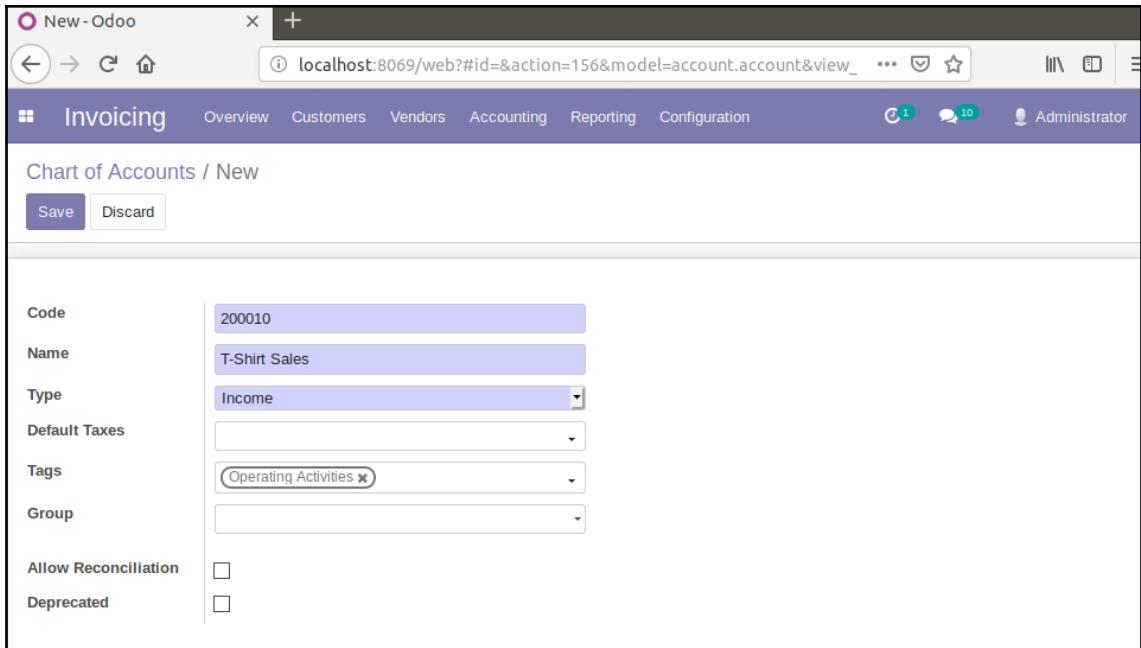
Practicing posting transactions and tracking the results

Remember that people spend many years and even get full university degrees in financial accounting. It is important that you take the time to learn how each process you implement affects the accounts in Odoo. When implementing an ERP system for your company, take the time to get this right. It will save you a lot of pain in the long run.

Setting up your own accounts

For default English installations, Odoo installs the standard **United States Chart of Accounts** template. Most companies, however, will need to modify this chart of accounts or even set up an entirely different chart of accounts to match the needs of their business. As an example, we are going to add an additional sales account specifically for T-shirts so that we can better organize our sales into types of products.

To set up a new account, go to the **Invoicing** menu and then down to the **Configuration** section, and under **Accounting**, choose **Chart of Accounts**. Odoo will present you with a list of all your current accounts in Odoo. Click on **Create** to add a new account:



The screenshot shows the Odoo web interface for creating a new account. The top navigation bar includes links for Invoicing, Overview, Customers, Vendors, Accounting, Reporting, Configuration, and Administrator. The main title is "Chart of Accounts / New". Below the title are two buttons: "Save" and "Discard". The form fields are as follows:

Code	200010
Name	T-Shirt Sales
Type	Income
Default Taxes	(dropdown menu)
Tags	Operating Activities
Group	(dropdown menu)
Allow Reconciliation	<input type="checkbox"/>
Deprecated	<input type="checkbox"/>

Note that in our screen, we have specified the account code as 200010. *Why did we choose this as the account code?* Odoo had already provided 200000 for the general **Product Sales** account; therefore, 200010 was an appropriate account code to choose for our **T-Shirt Sales**. For the name of the account, we have simply chosen **T-Shirt Sales**.

The other important setting is the **Account Type**. Odoo needs to know the type of account you are setting up. So, for example, if you were setting up an account that was to track the costs of products you must purchase to produce your products, you would specify an expense account type.



Take the time to plan your chart of accounts in Odoo. Even if your company has already been using an existing chart of accounts, it is always a good idea to evaluate the current chart of accounts and make any improvements, depending on the current state of the business.

Specifying a new account for your product category

With Odoo, you can manage accounts at the product category level. Therefore, all the products in a given category can utilize the same account settings. Let's create a new product category, T-shirts, for our Medium White T-Shirt, and assign that category to the 200010 T-Shirt Sales account we created. Later, we can add all T-shirt products to this category.

Go to the **Inventory** menu and, in the **Configuration** section, choose **Product Categories** from the **Products** submenu. This lists the current product categories:

The screenshot shows the Odoo web interface for managing product categories. The top navigation bar includes links for Overview, Operations, Master Data, Reporting, Configuration, and a user icon for Administrator. Below the navigation is a search bar labeled "Search..." and filter buttons for "Filters", "Group By", and "Favorites". A page number indicator shows "1-3 / 3". The main content area displays a list of categories under the heading "Product Category". The listed categories are: "All", "All / Expenses", and "All / Saleable". There are also buttons for "Create" and "Import".

At this point, you will see that we only have two categories. Click **Create** to create a new category for our T-shirt products:

Category name
T-Shirts

Parent Category
All / Saleable

Force Removal Strategy

Inventory Valuation

Costing Method
Standard Price

Inventory Valuation
Manual

Account Properties

Price Difference Account		Stock Input Account	101120 Stock Interim Account (Received)
Income Account	200010 T-Shirt Sales	Stock Output Account	101130 Stock Interim Account (Delivered)
Expense Account	220000 Expenses	Stock Valuation Account	101110 Stock Valuation Account

Account Stock Properties

Stock Journal	Stock Journal (USD)
---------------	---------------------

We have named our new category T-Shirts. All T-shirt products can now be grouped under this category. Also note that we have set a parent category of **All / Saleable**. This allows you to view the T-shirt products, along with all the other products, when you choose the parent category.

The most important thing from an accounting standpoint is the fact that we have assigned the **Income Account** 200010 T-shirt Sales, which we set up in the **Chart of Accounts**. When an invoice is posted that has a line item attributed to this product category, the amount for that line item will be posted to 200010 T-Shirt Sales. For the **Expense Account**, we have specified the built-in 220000 Expenses. This will post our expenses related to products in this category to that account.

Practicing with another product

Now, try going into the product record for Medium White T-Shirt and set the **Product Category** to **T-Shirts**. Create a sales order, turn it into an invoice, and validate it. By viewing the **Journal Entries**, you will see the income for your T-shirt in the specified income account.

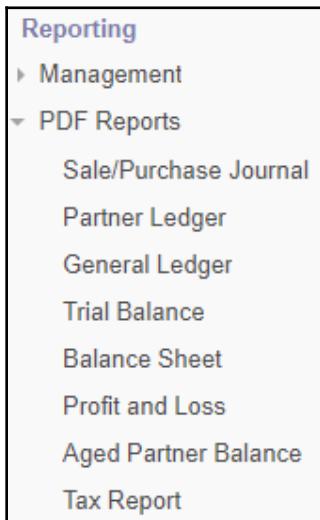
Remember that it is important to practice using Odoo until you are comfortable setting up accounts and understand where the transactions are posted. A little bit of time and effort put in during the configuration will save you a lot of time later.

Examining the available legal reports in Odoo

Like nearly all accounting and finance systems, Odoo provides the standard reports you would expect, including the following:

- General Ledger
- Trial Balance
- Balance Sheet
- Profit and Loss
- Financial Report

These reports are buried near the middle of the **Invoicing** menu in the **Reporting** section, and then accessed through **PDF Reports**:



Each report you select will bring up the corresponding wizard to specify the criteria for a given report. After you have made your selections and generated the report, you will be prompted to download the PDF file that contains the results. While going through each of these reports with all the screenshots is beyond the scope of this book, you are encouraged to spend some time examining each report and making sure that you understand how it fits within the reporting requirements for your business.



As you add more and more data to your system, some of the accounting reports will take additional time to process. As part of the testing process before you go into production, you should take the time to make sure that all your accounting reports run at acceptable levels of performance using data that will simulate real-world conditions.

Creating journal entries

While Odoo will create many journal entries automatically when you perform various operations in the system, it is inevitable that, at some point, you (or your accountant) will wish to create a manual journal entry. A manual journal entry allows you to adjust account balances in a way that can be easily tracked and audited.

For our example, we are going to create a journal entry that will account for a small investment by one of the company owners. When someone puts money as an investment into a company, they are not buying anything and they are not selling anything. While there are other potential methods that can be used to reflect an investment, a simple journal entry is a straightforward way to accurately record the transaction.

To enter a journal entry, go to the **Invoicing** menu, select **Journal Entries** in the **Journal Entries** section, and click the **Create** button:

The screenshot shows the Odoo interface for creating a new journal entry. The top navigation bar includes links for Discuss, Calendar, Contacts, CRM, Sales, Purchases, Link Tracker, Inventory, Manufacturing, Invoicing (which is selected), Apps, and Settings. The title bar says "Journal Entries / * 6". The main area displays a table for a new journal item. The table has columns for Account, Partner, Label, Debit, and Credit. Two entries are listed: "300100 Capital" with a debit of \$0.00 and a credit of \$5,000.00, and "101401 Bank" with a debit of \$5,000.00 and a credit of \$0.00. The journal is set to "Bank (USD)". The date is 12/15/2017 and the reference is "Owner Investment". A "Reconciled entries" button is visible in the top right corner of the table area.

When you first create a new journal entry, you will need to pick which journal to post to. For this situation, we used an example of how we can post to the **Bank** journal to represent the owner's investment into the company.

We are considering the \$5,000 investment a cash investment by the owner, so we have posted this into the **Bank** journal.

Whenever you create a journal entry, you will add at least two line items. Furthermore, the line items must balance out. In our example, we are putting \$5,000 in funds from the owner into the company bank account.

Typically, any investment the owner puts into the company must also be recognized as a liability for the company. *Why?* The money really doesn't belong to the company. Instead, the money, which in our case is \$5,000, is considered the owner's equity. The owner is entitled to get that money back, and therefore it is booked as a liability. You can verify this by opening up the chart of accounts and looking at the list of main accounts. **Liabilities and Equity** are grouped together and are then divided out as you drill down into the account hierarchy.

Odoo sets up a **Capital** account that allows us to post the \$5,000 we have put into the bank as capital stock for the owner. Once you save your journal entry, it is in draft form. To post the journal entry and have it appear in your reports, you must click the **Post** button.

Summary

In this chapter, we examined how Odoo generates transactions and how you can use the chart of accounts to look at how those transactions originated. We examined both the **Accounts Payable** and **Accounts Receivable** accounts and looked at how an invoice is posted. There are certainly more advanced Odoo topics, such as bank reconciliation and recurring entries, that are beyond the scope of this book. Please refer to the [Appendix A, Locating Additional Odoo Resources](#), to locate additional resources on more advanced Odoo subjects.

In the next chapter, we will take a closer look at some important topics to consider when administering an Odoo installation.

7

Administering an Odoo Installation

One of the greatest advantages of Odoo is that it is easy to get up and running with very little setup. Within just a few minutes, you can have several applications installed and you can begin working with the system right away. In the previous chapters, we covered a great deal of functionality without having to spend a lot of time on configuration, access rules, languages, or other administrative topics.

Now, we will take a closer look at some important topics to consider when administering an Odoo installation.

In this chapter, we will cover the following topics:

- Strategies for server setups, administration, and backups
- Creating users and groups and assigning access rights
- Internationalization, including language translation and currencies
- How to manage document sequences
- Multi-company configurations

Basic considerations for an Odoo administration

Like most IT installations, successful Odoo installations require proper planning and maintenance. Care must be taken in documenting important configuration details, and you must always have a business continuity plan in place that focuses on getting your Odoo installation back up and running within an acceptable period of time.

Have an implementation strategy

While you are learning about Odoo and prototyping how you may use it for your business, you may not care much about a clear implementation strategy; however, once you have made the decision to use Odoo for your business, it is important to plan your implementation strategy. While you may not have the time to write out a 150-page detailed strategy, it is important to take the time to document your overall strategy and have a plan in place before you begin setting up servers and installing Odoo.

While the total breadth of project management and administration that goes into an ERP system is beyond the scope of this book, there are several basic implementation considerations you will always want to think about.

Development, staging, and production servers

One of the first considerations you will need to keep in mind when contemplating an Odoo installation is how you will configure servers for various Odoo instances that may be required during planning, deployment, and final production operations. For example, you don't want to be making modifications to Odoo's functionality in your live production system. Instead, you should always make changes and modifications in a development instance of Odoo, where you can test your changes outside of the live database.

In addition to a development server and a production server, it is often desirable to have an Odoo installation that users (and, in some cases, business partners) can use to learn the operations of the system. Sometimes, this installation is known as the staging server. This server will typically have all the tested changes and functionality of the live system, but will be loaded with test data and configurations that are useful for training.

Each installation will have its own requirements and constraints. What is important is that you make these decisions early on in your Odoo configurations so that you can properly administer the installations all the way from development to production.

Clear documentation of all Odoo configurations

Once you have decided what Odoo servers you require and how those installations should be configured, it is important that you create a clearly defined method for documenting all the details that go along with the setup. This can be as simple as a text or Microsoft Word document that is in a known place and kept up to date, or it can be as complex as a full-blown project manager. Using cloud organization tools such as Dropbox, Evernote, and Google Documents provides you with a lot of options for how you can document your Odoo installations.

It will be up to your own business policies to determine exactly where you store this information and how much detail you keep. It is important to note, however, that it is almost always better to err on the side of having too much detail rather than too little. You will need to be aware of how you secure usernames and passwords and have a clear policy on how that information is securely stored.

Focusing on business continuity

Any business information system is only as good as its ability to recover from something going wrong. Despite having more reliable hardware and software, data can still get corrupted. Even the most dedicated employee can accidentally post bad data. The best security can be defeated. Despite Odoo's best efforts, there are probably bugs still lurking in its applications. No amount of planning can prevent a problem from occurring. This is why one of the most important tasks when administering an Odoo installation is making sure that you always have a clear recovery strategy.

Here are a few important considerations to remember:

- Regularly test your backups for recoverability. Just because you are backing up your data doesn't mean that it is quickly recoverable. All too often, businesses can go months—or even years—without testing whether the data they are backing up is recoverable.
- Have a strong archive of backups. Perform daily backups along with weekly and monthly snapshots as well. Often, data can be bad, deleted, or corrupted long before anyone knows anything has gone wrong. Someone might accidentally delete a set of old entries, and it may not be until a few months later that a manager doing a report finds critical holes in the data.

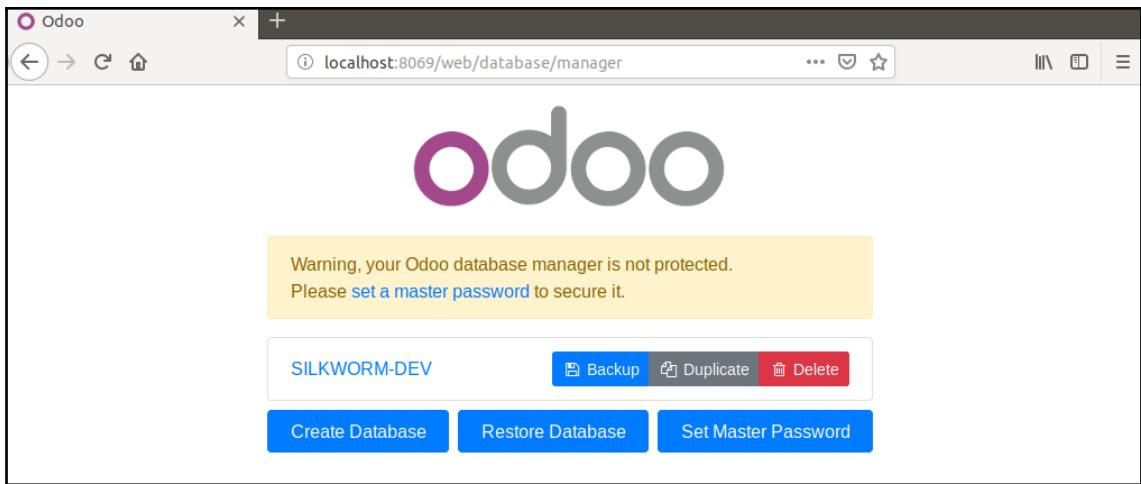
- Have contingency system options. Even if you plan on hosting locally, consider having a cloud server configured where you can run your Odoo installation in a pinch. Too often, you can have the backups ready to go but you are still waiting on hardware to be fixed or a part to be delivered, which is going to extend your downtime. If you plan to use your development server as a backup production system, make sure that you have the proper procedures in place and tested. Don't make optimistic assumptions about your system contingencies. Test them at least once or twice a year.
- Make sure that you know how long it takes to fully recover your Odoo installation and what data would need to be re-entered into the system. If you back up nightly, and it takes you four hours to get your installation back up and running, make sure that your internal business processes are clear on exactly what steps are required.
- Know exactly how much downtime costs your business and plan accordingly. Companies such as eBay and Amazon are in crisis if they are down for even a few minutes; more than an hour of downtime for them would make international news. While you may not have their uptime requirements, it is important that you understand exactly what risks your business faces if your Odoo installation goes down for two minutes, two hours, or even two days.

Backing up your Odoo database

It is critical in a production environment that, at a minimum, you back up both your working Odoo application directories and the associated Postgres databases. Ideally, you will have server snapshots and a clear business continuity plan in place and tested. Still, it is valuable to know that Odoo provides a built-in database backup tool. I use it frequently in a variety of Odoo installations.

Before going ahead, it should be noted that this function will not be applicable to all Odoo installations. If you are running in a hosted Odoo environment where you have been provided with login credentials for your database, then you will be provided with a specific backup procedure. Make sure that you fully understand how it works and have a way to test and make sure that it functions as expected.

The easiest way to get to the backup database function is to navigate directly to the database manager by adding `/web/database/manager` to the end of your Odoo URL. This will result in the following page appearing:

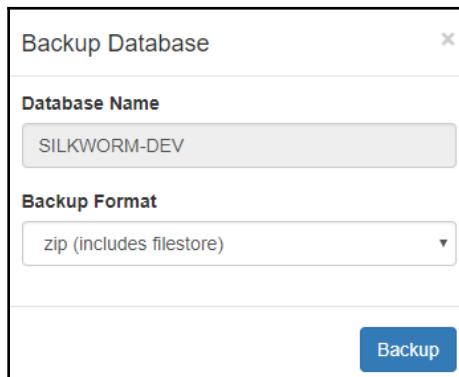


Here, you will see all the databases in your Odoo installation, along with the options to **Backup**, **Duplicate**, or **Delete** each of the databases.



Sometimes, if you are having trouble with an Odoo installation, such as getting **Internal Server** errors and other system-related issues, you can navigate directly to the database manager to back up your database and perform operations that may help you recover from the problem.

Clicking the **Backup** button will bring up a simple wizard that will allow you to back up the database in a `.zip` format that will contain all the filestores associated with that database, or as a `pg_dump` file. A filestore could contain things such as document attachments or pictures associated with your products. The `pg_dump` format is a standard Postgres database operation that will generate a backup of the database, except without the associated filestore. In the following screenshot, we have chosen the `.zip` format:



Click the **Backup** button to begin backing up the database. The database will then begin downloading. (If your database is extremely large, there is a chance that the file may not be easily downloaded.)

Restoring an Odoo database

The ability to back up a database does little good if you don't have the ability to restore the database and get it back up and running. You can click the **Restore Database** button to bring up the **Restore Database** form, as follows:

The screenshot shows a modal dialog titled "Restore Database". It has a "File" section with a "Choose File" button containing the text "SILKWORM-D...-36-21.zip". Below it is a "Database Name" input field. A note below the input field states: "This database might have been moved or copied. In order to avoid conflicts between databases, Odoo needs to know if this database was moved or copied. If you don't know, answer 'This database is a copy'." There are two radio buttons: one selected for "This database is a copy" and another for "This database was moved". At the bottom right is a blue "Continue" button.

Here, you specify the **Master Password** and choose the file you wish to restore. Once you have selected the file, you will need to specify a new database name to restore the database into it.

You also have the option to choose either **Backup Restore** or **Copy of an existing database** as the **Mode** for restoration. As the instructions explain, Odoo will handle the restore slightly differently if you are restoring a database that has been moved rather than copied. A moved database would be one in which you don't intend to have another instance running, whereas a copied database would assume that the other database stays in use. Use the appropriate mode for your situation.

Administering users in Odoo

In any ERP system, it is important that you completely understand how users and user access rights are managed. When Odoo is first installed, an admin account is created automatically. This is a superuser account, and it is the only one like it. In some systems, any account can be specified to have full administrative privileges. Odoo, however, gives permissions to the administration account that no other user in the system has.

Specifically, all access rights are bypassed when using the administrator account. Much like the root account in Linux or Ubuntu, you always need to protect your administration account by using a strong password and keeping it secret.

Selecting a user to administer

Let's begin by looking at a user in Odoo and see how they are tied to partner records within Odoo applications.

To access the list of users, click **Settings** in the main menu under **Users & Companies** and choose **Users** from the **Users** section in the left-hand menu, as follows:

We can now click on **Terry Zeigler** to bring up the user. By doing this, we can look at the additional options that are available:

The screenshot shows the Odoo web interface for managing users. The URL in the browser is `localhost:8069/web?db=SILKWORM-DEV#id=6&action=68&model=res.users&view_type=list`. The top navigation bar includes links for Settings, Dashboard, Users & Companies, Translations, General Settings, and a user icon for Administrator. Below the header, there are buttons for Edit and Create, and a dropdown for Action. A progress bar indicates 2 / 2 steps completed. The main content area displays a user record for "Terry Zeigler" with the email "terryzeigler.exampleemail.com". A status indicator shows "Active" with a green checkmark. Below the profile, there are tabs for Access Rights and Preferences. The "Application Accesses" section lists various Odoo modules: Sales (Manager), Inventory, Manufacturing, Accounting & Finance (Billing Manager), Purchases, and Administration. The "Other" section includes checkboxes for Access to Private Addresses (unchecked), Quotation Templates (unchecked), Tax display B2B (checked), and Tax display B2C (unchecked).

When you pull up a user in your own Odoo installation, it is unlikely that your screen will look exactly like the preceding screenshot. Depending on the exact applications that are installed, the available application accesses will change. It is also common to see specific modifications to this screen, depending on the Odoo build you are running.

Managing user preferences

Now, click on the **Preferences** tab and take a look at the available options:

The screenshot shows the Odoo web interface with the title bar "Terry Zeigler - Odoo". The main navigation bar includes "Settings", "Dashboard", "Users & Companies", "Translations", and "General Settings". On the right, there are notifications for 1 message and 10 messages, and a dropdown for "Administrator". The current view is "Users / Terry Zeigler". At the top left, there are "Edit" and "Create" buttons, and an "Action" dropdown. A progress bar at the top right indicates "Never Connected" to "Confirmed". Below this, the user profile for "Terry Zeigler" is displayed, showing the email "terryzeigler@examplemail.com" and an "Active" status with a checkmark. There are tabs for "Access Rights" and "Preferences", with "Preferences" currently selected. The "Localization" section shows "Language" set to "English" and "Timezone" listed. The "Messaging and Social" section shows "Notification Management" set to "Handle by Emails" and includes "Alias" and "Signature" options.

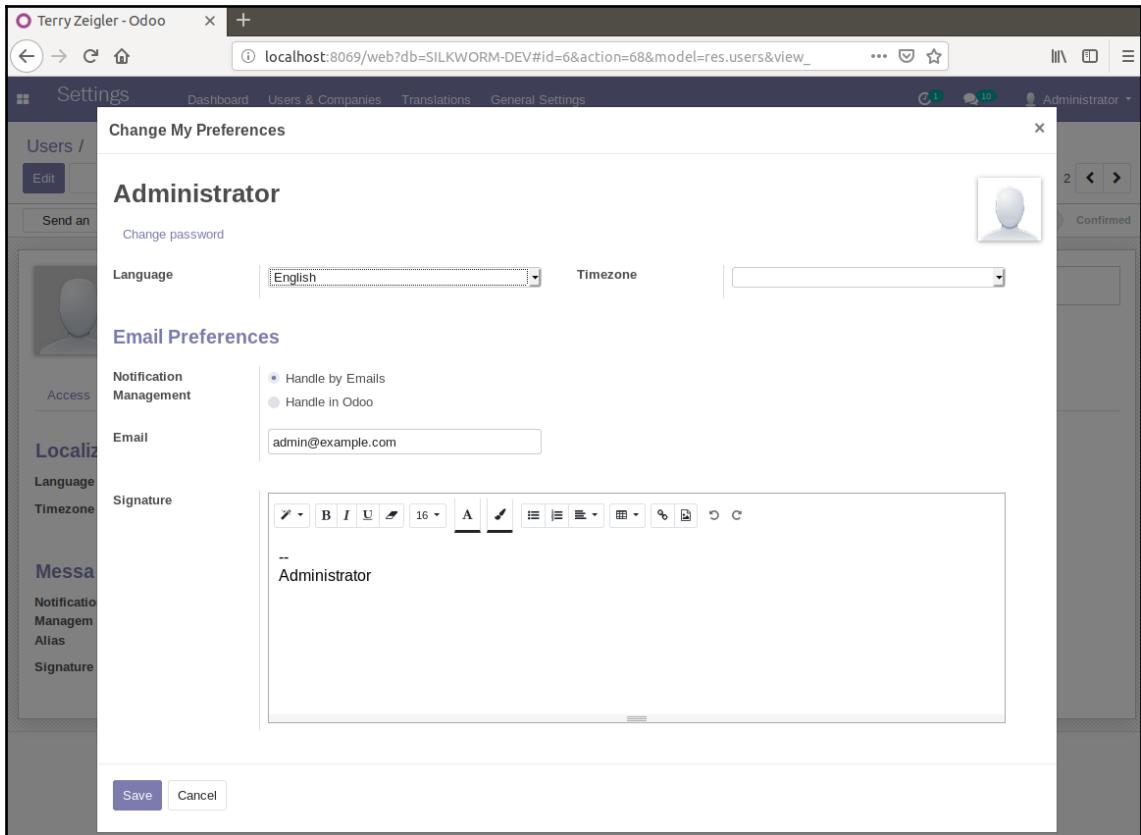
In this section, we can specify important localization options that can dramatically change the user experience. We can specify one of the many languages that Odoo supports, as well as the time zone and default sales team of the user.

Additionally, the **Preferences** section lets you manage your **Messaging and Social** options for a user.

Currently, there are only two options for receiving inbox notifications by email—either a user never receives notifications or receives all of the notifications.

The **Alias** option will allow administrators to configure an email alias for the user. By creating an alias, the user can receive incoming messages from an email that is different to the one that's been assigned to the account.

Finally, you can use the **Signature** rich text area at the bottom of the page to specify a signature footer for emails that are sent by this user. If desired, the user can change their own signature at any time by choosing **Preferences** from the menu in the upper-right corner of the screen, as shown in the following screenshot:



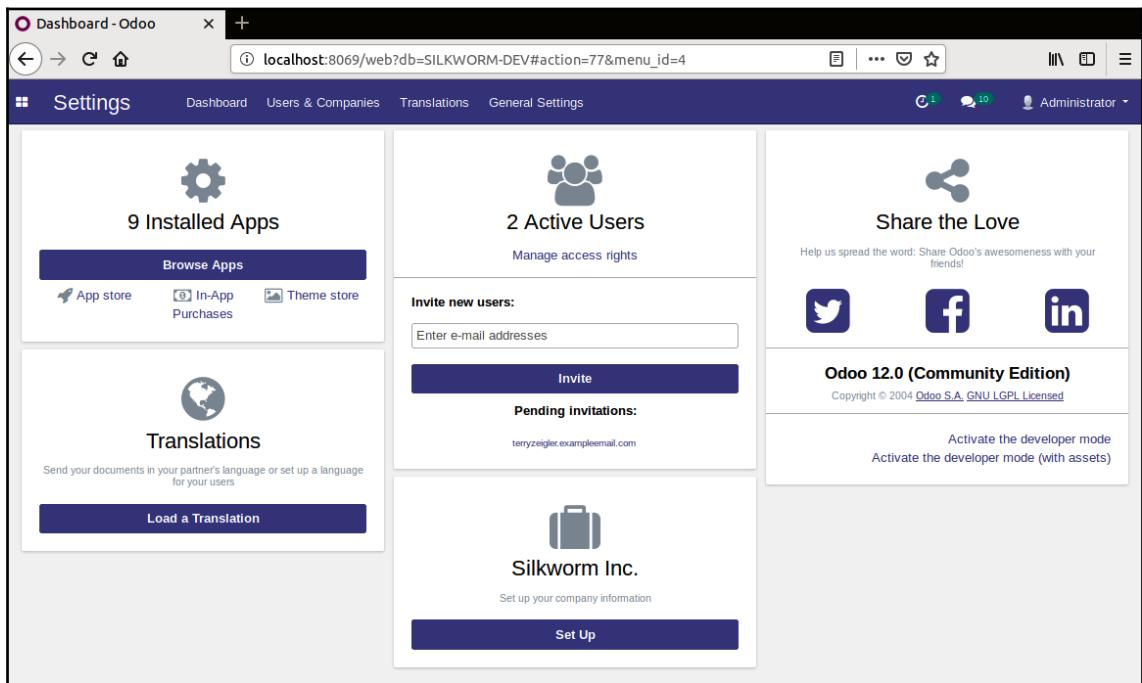
Remember that if you are an administrator making these changes for another user, changing these settings will require that user to log out of the system before their session can be updated with the changes.



If you are setting up a lot of users, don't forget that you can use the **Duplicate** option under the **More** menu at the top of the form to make a copy of a user. This can be handy if a worker has left and has been replaced with another worker. You can deactivate the old employee and duplicate their profile for the new employee.

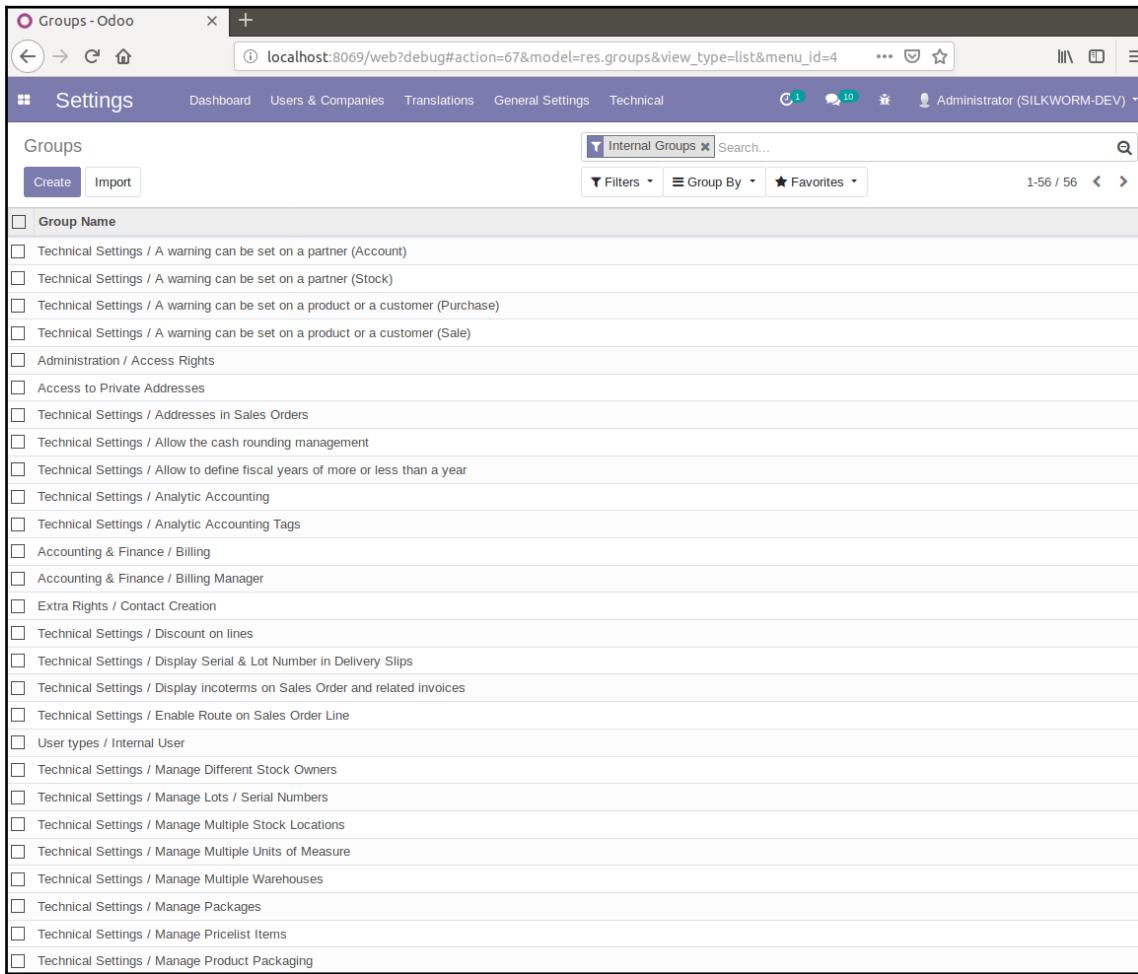
Understanding groups in Odoo

In Odoo, you give users permissions by assigning the users to groups. Once a user is assigned to a group, the user has all the permissions and options that are associated with that group. Users can belong, and often do belong, to more than one group. To see the list of groups that are currently available in your Odoo installation, you will need to turn on the developer settings. To do this, go to the **Settings** menu at the top of the page and then choose **Dashboard**, as shown in the following screenshot:



On the right-hand side of the page, you will find a link to **Activate the developer mode**. Clicking this link will then refresh the **Settings** menu and provide you with many more options for administering your Odoo installation.

Once you have successfully entered developer mode, you can manage the user groups in Odoo by clicking the **Groups** link under the **Users & Companies** menu:



The screenshot shows the Odoo web interface with the title "Groups - Odoo". The URL in the address bar is "localhost:8069/web?debug#action=67&model=res.groups&view_type=list&menu_id=4". The top navigation bar includes links for Dashboard, Users & Companies, Translations, General Settings, Technical, and a dropdown for the Administrator (SILKWORM-DEV). Below the header is a search bar with the placeholder "Internal Groups" and a filter section with "Filters", "Group By", and "Favorites". A toolbar on the right provides navigation controls. The main content area is titled "Groups" and contains a table with a single column of checkboxes and group names. The list is very long, starting with "Technical Settings / A warning can be set on a partner (Account)" and ending with "Technical Settings / Manage Product Packaging".

As you can see, Odoo has a lot of groups. Fortunately, once you understand how groups work, you will be able to easily determine exactly what options are available to a user when you put them in a specific group.

For our example, let's examine the **Sales/Manager** group. You can find this group by scrolling down the list of groups or by using the search function to narrow down the list until you find the group you are looking for.

Like other lists, clicking the **Sales / Manager** group brings up the following form:

The screenshot shows the Odoo web interface for managing groups. At the top, there's a header bar with the title "Sales / Manager - Odoo" and a URL "localhost:8069/web?debug#id=21&action=67&model=res.groups&view_type=form". Below the header is a navigation bar with links like "Settings", "Dashboard", "Users & Companies", "Translations", "General Settings", "Technical", and a user "Administrator (SILKWORM-DEV)". On the right side of the header, there are icons for notifications (10), messages, and other system status.

The main content area is titled "Groups / Sales / Manager". It has tabs for "Edit" (which is selected) and "Create". There's also a "Action" dropdown and a page number "35 / 56" with navigation arrows.

The main table displays the group details:

Application	Sales	Name	Manager
Share Group	<input type="checkbox"/>		

Below the table, there are tabs for "Users", "Inherited", "Menus", "Views", "Access Rights", "Record Rules", and "Notes". The "Users" tab is selected, showing a list of users assigned to the group:

Name	Login	Language	Latest connection
Administrator	admin	English	01/11/2019 18:39:54
Terry Zeigler	terryzeigler.exampleemail.com	English	

At the very top of the form on the left, you can see that a group is always associated with a given application. In this case, the group is associated with the **Sales** application. On the right is the name, which in our case is **Manager**. Odoo automatically adds a slash (/) between the application and the name when displaying a full name in the list.

As you can see, the first page lists the users that are assigned to the group. Naturally, you can add and remove users to and from this group as required. You will also notice that there are seven pages on this form that allow you to configure exactly what permissions a group will offer to its users.

Understanding group inheritance in Odoo

Managing access permissions in any ERP system is always a challenge. Odoo makes managing user permissions a little easier by allowing you to inherit permissions from multiple groups and then define a new group that automatically includes all the permissions from those groups. With proper planning, this allows you to create groups that provide your users with the permissions they require.

Let's take a look at the groups that are **Inherited** by the **Sales / Manager** group:

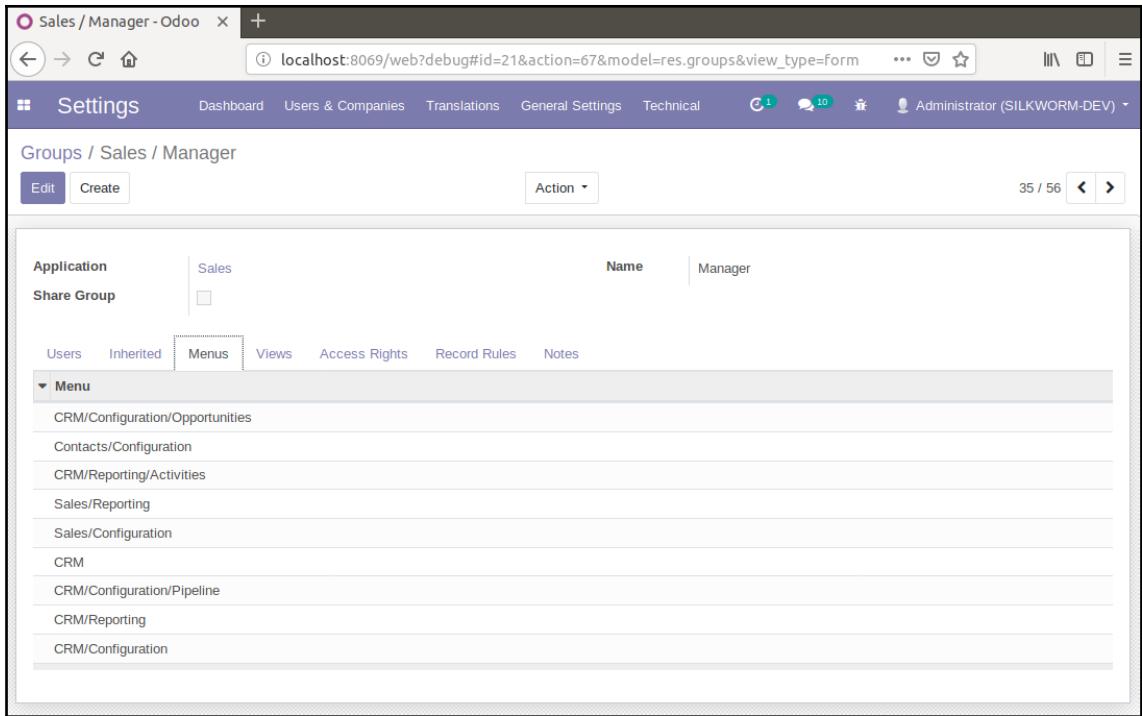
The screenshot shows the Odoo web interface with the title bar "Sales / Manager - Odoo". The URL in the address bar is "localhost:8069/web?debug#id=21&action=67&model=res.groups&view_type=form". The top navigation bar includes "Settings", "Dashboard", "Users & Companies", "Translations", "General Settings", "Technical", and a user dropdown for "Administrator (SILKWORM-DEV)". Below the header is a toolbar with icons for creating, deleting, and filtering records, along with a page number "35 / 56". The main content area is titled "Groups / Sales / Manager". It has tabs for "Edit", "Create", and "Action". Under the "Users" tab, the "Inherited" tab is selected. A note below says "Users added to this group are automatically added in the following groups." A "Group Name" section lists "Sales / User: All Documents". Other tabs include "Menus", "Views", "Access Rights", "Record Rules", and "Notes".

The **Sales / Manager** group has **Sales / User: All Documents** included in the inherited list. Just like the instructions say, users that are added to the **Sales / Manager** group will be automatically added to the **Sales / User: All Documents** group.

Manager groups such as this will often include all the other groups that have more restrictions in the system. For example, looking at the **Sales / User: All Documents** group allows you to see the most restrictive group permissions for the sales group.

Defining menus for your group

Groups provide a direct way of determining what menus users in that group have access to. In the case of the **Sales / Manager** group, we have additional menu options listed, but members of the **Sales / User: All Documents** group wouldn't see these menus unless they are also members of **Sales / Manager**, or if they are added specifically to the **Sales / User: All Documents** group:



If, for example, you wanted to allow users in the **Sales / User: All Documents** group to view the **Activities** report, you could remove the menu from the list in the **Manager** group and add the menu to the **Sales / User: All Documents** group. Because the **Manager** group inherits from **Sales / User: All Documents**, its users will still be able to see the menu—as will those who are only in the **Sales / User: All Documents** group.

Understanding access rights in Odoo

So far, we have seen how groups can inherit from other groups and how menus can be assigned to a specific group. Now, we will look at what access rights determine which models a group has access to and what permissions they are assigned.

In the following screenshot, you can see the **Access Rights** page for the **Sales / Manager** group:

The screenshot shows the Odoo web interface for managing access rights. The title bar says "Sales / Manager - Odoo". The URL is "localhost:8069/web?debug#id=21&action=67&model=res.groups&view_type=form". The top navigation bar includes "Settings", "Dashboard", "Users & Companies", "Translations", "General Settings", "Technical", and "Administrator (SILKWORM-DEV)". Below the navigation is a toolbar with icons for "C 1", "T 10", "W 1", and "M 1". The main content area is titled "Groups / Sales / Manager" and shows the "Manager" group details. It has tabs for "Edit", "Create", "Action", "Users", "Inherited", "Menus", "Views", "Access Rights" (which is selected), "Record Rules", and "Notes". A note at the bottom left says "1-40 / 41". The "Access Rights" table has columns: Name, Object, Read Access, Write Access, Create Access, and Delete Access. The table lists various Odoo models and their access rights for the "Manager" group.

Name	Object	Read Access	Write Access	Create Access	Delete Access
crm.team.manager	Sales Channels	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
account_invoice.manager	Invoice	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sale.order.manager	Sale Order	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
sale.report	Sales Analysis Report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
res.partner.sale.manager	Contact	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
uom.category.salemanager	Product UoM Categories	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
uom.uom.salemanager	Product Unit of Measure	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.category.salemanager	Product Category	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.supplierinfo.salemanager	Supplier Pricelist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.pricelist.salemanager	Pricelist	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
res_partner_group_sale_manager	Contact	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
product.pricelist.item.salemanager	Pricelist Item	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
prices.history.sale.manager	Product Price List History	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.template.salemanager	Product Template	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.product.salemanager	Product	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.attribute.manager	Product Attribute	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.attribute.manager.value	Attribute Value	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.template.attribute.manager.value	Product Attribute Value	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.attribute.manager.filter.line	Product Template Attribute Exclusion	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product.attribute.manager.line	Product Template Attribute Line	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Access Rights is where you define exactly what models the group has access to. In Odoo, the term **Model** represents a business entity object and its related operations. You can determine whether a group has any combination of **Read Access**, **Write Access**, **Create Access**, and **Delete Access**, and also do this for each group. For example, in the preceding list, if you scroll down, you will find that the **Sales / Manager** group has the ability to read, write, and create meeting types, but they cannot delete meeting types.

Now, let's take a quick look at the access rights of the **Sales / User: Own Documents Only** group:

The screenshot shows the Odoo web interface with a dark theme. The top navigation bar includes links for Dashboard, Users & Companies, Translations, General Settings, and Technical. A user named 'Administrator (SBUKORM-DEV)' is logged in. On the left, a sidebar lists 'Groups / Applications / Share Groups' under 'Sales / User'. The main content area is titled 'Open: Inherits' and shows the 'Access Rights' tab selected for the 'Sales' application and 'User: Own Documents Only' group. The table displays access permissions for various objects. The columns are: Name, Object, Read Access, Write Access, Create Access, and Delete Access. Most objects have 'Read Access' checked, while 'Write Access' is mostly unchecked except for Sale Order, Sales Order Line, and Pricelist. 'Create Access' is checked for Sale Order, Sales Order Line, and Pricelist. 'Delete Access' is checked for Company Property, Pricelist, and Account.

Name	Object	Read Access	Write Access	Create Access	Delete Access
crm.team.user	Sales Channels	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sale.order	Sale Order	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
sale.order.line	Sales Order Line	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
account_invoice_tax_salesman	Invoice Tax	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
account_invoice_salesman	Invoice	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
account_invoice_line_salesman	Invoice Line	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
account_payment_term_salesman	Payment Terms	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
account.analytic.tag.sale.salesman	Analytic Tags	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
account_analytic_account_salesman	Analytic Account	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
sale.report	Sales Analysis Report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ir.property.sales	Company Property	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
account.journal.sale.order.user	Journal	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
res.partner.sale.user	Contact	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
product.template.sale.use	Product Template	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
product.product.sale.use	Product	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
account.tax.user	Tax	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
uom.uom.user	Product Unit of Measure	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
product.pricelist.sale.user	Pricelist	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
account.account.salesman	Account	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

The **Manager** group we looked at previously allowed users to create and write records, but the **Sales / User: Own Documents Only** group has read access for many objects, such as **Sales Team, Journal, Partner, Product Template**, and so on. This group can see the necessary information, and it can be selected on forms and reports, but its members don't have permission to modify records in those objects.

Understanding group rules in Odoo

Sometimes, in a system, you want users to have access to a particular model, but not to all the records in that model. For example, you may want users to have access to phone calls within the system, but only to records of their own phone calls rather than all the calls from the whole system. When you need to control user access based on the contents of records within a model, you can define **Rules**.

For this example, we are looking at the rules for the **Sales / User: Own Documents Only** group. Because this is a very restricted group, there are many rules that limit users in this group, meaning that they only see records that are associated with them personally:

The screenshot shows the Odoo Settings interface with the 'Groups' tab selected. A modal window titled 'Open: Inherits' is displayed, showing the 'Record Rules' tab for the 'Sales' application. The table lists various record rules with checkboxes for 'Apply for Read', 'Apply for Write', 'Apply for Create', and 'Apply for Delete'. Most rules have all four checkboxes checked.

Name	Object	Domain	Apply for Read	Apply for Write	Apply for Create	Apply for Delete
All Orders Lines	Sales Order Line	[(1,'=',1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
All Orders Analysis	Sales Analysis Report	[(1,'=',1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
All Invoices Analysis	Invoices Statistics	[(1,'=', 1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
All Activities	CRM Activity Analysis	[(1,'=',1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
All Salesteam	Sales Channels	[('team_type','in', ['sales', 'website'])]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
All Orders	Sale Order	[(1,'=',1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
All Leads	Lead/Opportunity	[(1,'=',1)]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Close

Typically, **Manager** groups will have little or no rules because they don't have restrictions on what records they can access. Groups such as **Own Documents Only** have quite a few rules so that users can't see records that don't belong to them. Let's take a quick look at the **Personal Orders** rule so that we can see how to construct a rule that limits what records a user can access:

The screenshot shows the Odoo web interface with a dark theme. The URL in the browser is `localhost:8069/web?debug#id=21&action=67&model=res.groups&view_type=form`. The main menu on the left includes 'Dashboard', 'Users & Companies', 'Translations', 'General Settings', 'Technical', and 'Administrative (SILKWORM-DEV)'. The central window is titled 'Open: Rules' and displays the configuration for the 'Personal Orders' rule.

General section:

- Name: Personal Orders
- Object: Sale Order

Access Rights section:

- Apply for Read:
- Apply for Create:
- Apply for Write:
- Apply for Delete:

Rule Definition (Domain Filter)

```
[!,('user_id','!=',user.id),('user_id','=',False)]
```

Groups (no group = global)

Global

Group Name: Sales / User: Own Documents Only

Interaction between rules

Global rules (non group-specific) are restrictions, and cannot be bypassed. Group-specific rules grant additional permissions, but are constrained within the bounds of global ones. The first group rules restrict further the global rules, but can be relaxed by additional group rules.

Detailed algorithm:

1. Global rules are combined together with a logical AND operator, and with the result of the following steps
2. Group-specific rules are combined together with a logical OR operator
3. If user belongs to several groups, the results from step 2 are combined with logical OR operator

Example: GLOBAL_RULE_1 AND GLOBAL_RULE_2 AND ((GROUP_A_RULE_1 OR GROUP_A_RULE_2) OR (GROUP_B_RULE_1 OR GROUP_B_RULE_2))

Close

Odoo provides a pretty good description at the bottom of the form on how rules interact. If no groups are specified in the list, this means that this rule will apply to everyone—all groups. As you can see in the preceding screenshot, you can specify the access rights for this rule. This means that you could have a rule in which a user can access (read) certain records, but they wouldn't be able to create, write, or delete records.

The most important part of the rule is the **Rule Definition** or **Domain Filter**. This is the filter that is applied to each record to determine whether that record should be available. While the syntax may look a bit cryptic, you can see that the system is checking that `user_id` is equal to the current `user_id`. Specifically, this filter will be true if you are looking at your own records, or records that have not been assigned to any specific user.



When making your own rules, copy and paste rules from a similar rule to make it easier to get the syntax right. Also, be careful about changing rules in a live system. It is possible that an error in your syntax could make it impossible to access certain parts of the system.

Internationalization in Odoo

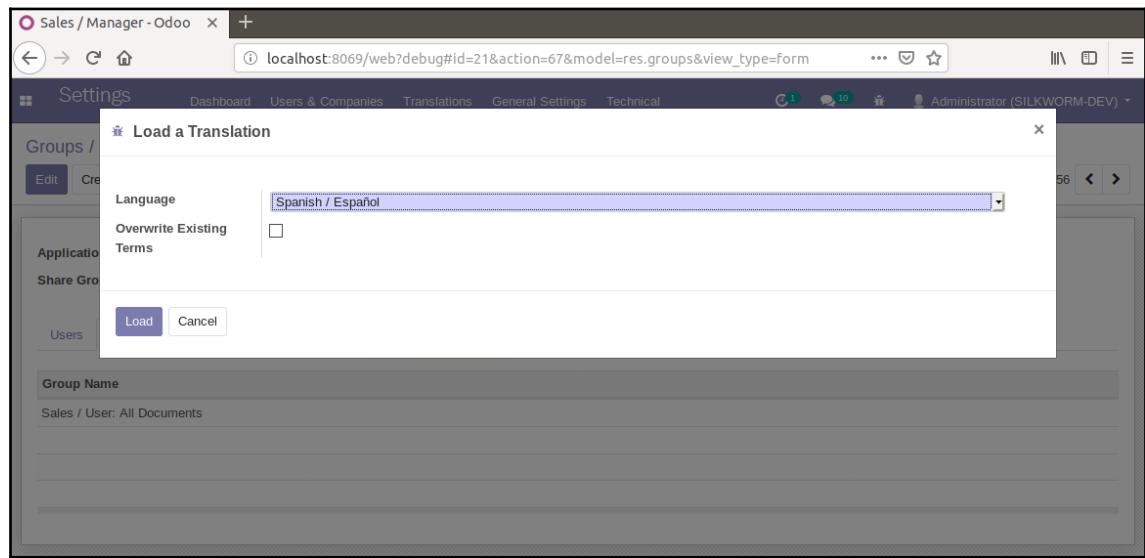
Even with a conventional English installation of Odoo, it is possible to configure Odoo to work with a variety of languages, time zones, and currencies without downloading any additional add-ons. Odoo has very robust features for configuring a global ERP system that can meet the demands of today's multicultural business environment.

Like most Odoo features, you only need to configure the international features you require for your business. For example, you may do business entirely in US dollars but would like to offer Odoo in Spanish for some of your workstations, users, or portal customers. On the other hand, if you are purchasing from a supplier in an alternative currency, you may choose to create a special price list that allows you to do business in that currency.

Configuring language translation

Like many of the other options in Odoo we have discussed, business requirements should drive how you configure your system. For our real-world example, we want to be able to offer a native Spanish Odoo interface for some employees. Let's see how we can configure Odoo to provide other language alternatives.

Fortunately, Odoo makes this very easy. Simply go to the **Settings** menu and choose **Load a Translation** in the **Translations** section, as shown in the following screenshot:



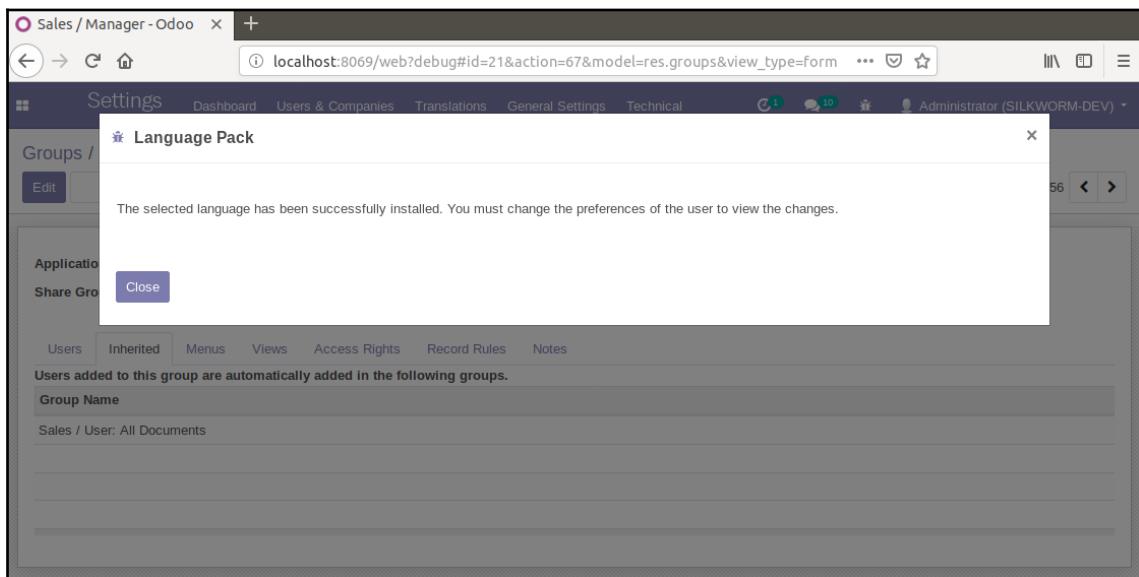
You will find quite a few languages to choose from in this list. At the time of writing, there are more than 80 languages to choose from.



Many of these languages are community-supported, and translation will certainly vary. Furthermore, ERP systems can often be confusing, even for users that speak the language fluently. Take the time to train users so that they understand all their processes well.

You would use **Overwrite Existing Terms** if you have made custom modifications to a language translation and now wish to overwrite them.

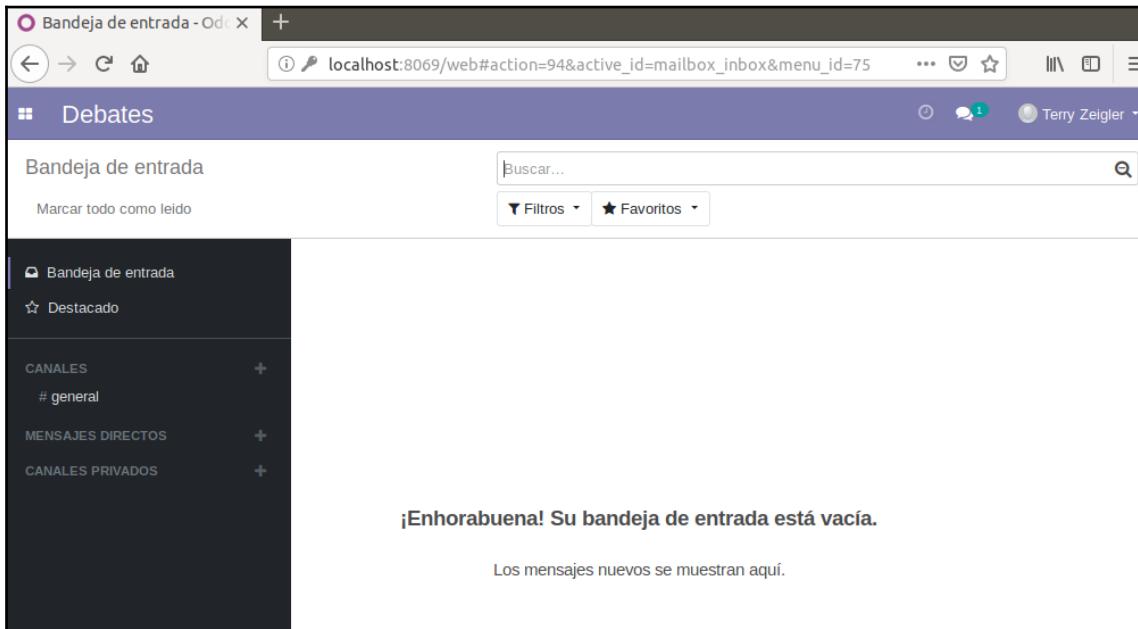
After the language is loaded, you will get a confirmation message and instructions on how to begin using the installed language, as follows:



Now that we have installed the new language, we can assign that **Language** to our users, customers, and even our suppliers. In the following screenshot, we can see that we have set the **Language** choice of user **Terry Zeigler** to **Spanish / Español**. Odoo also allows you to specify the **Timezone** by either the GMT offset or by common regions. In this case, we have chosen **America/Cancun** as the **Timezone**:

The screenshot shows the Odoo web interface for managing users. The top navigation bar includes links for Dashboard, Users & Companies, Translations, General Settings, Technical, and an Administrator dropdown. The main content area is titled "Users / Terry Zeigler". It displays a user card for "Terry Zeigler" with the email "terryzeigler.exampleemail.com" and a status indicator showing "Active". Below the card, there are tabs for "Access Rights" and "Preferences". The "Localization" section shows "Language" set to "Spanish / Español" and "Timezone" set to "America/Cancun". The "Menus Customization" section shows "Home Action". The "Messaging and Social" section lists "Notification Management", "Alias", "OdooBot Status" (set to "Not initialized"), and "Signature".

After the changes have been saved and the user has logged back in, we will see that their interface has changed to Spanish:



Users may also change their own language and time zone settings at any time by clicking on their username in the upper-right corner and choosing **Preferences** from the menu. As long as the language translation has been loaded into the system, it remains available to all users.

International currencies

As we've already seen, it is quite easy to configure Odoo for multiple languages. **Currencies**, however, will require more planning and testing during system configuration. Unlike languages, multiple currencies can directly modify the amount of money you are receiving or paying out. If the system has misconfigured currency settings, you are almost guaranteed to have inaccurate transactions within your system at some point. Make sure that you thoroughly test all possible scenarios when you're working with multiple currencies in Odoo or any other ERP system.

To set up **Multi-Currency** in Odoo, go to the **General Settings** menu and select **Invoicing**, as shown in the following screenshot:

The screenshot shows the Odoo web interface for 'New - Odoo' at the top. Below it, the main navigation bar includes 'Dashboard', 'Users & Companies', 'Translations', 'General Settings', 'Technical', and a user dropdown for 'Administrator (SILKWORM-DEV)'. On the left, a sidebar lists modules: CRM, Sales, Purchase, Inventory, Manufacturing, Invoicing (which is selected and highlighted in orange), and General Settings. The main content area is titled 'Settings' and contains several sections: 'Setup' (Bank Reconciliation Threshold), 'Taxes' (Default Taxes for Sales and Purchase Tax set to 'Tax 15.00%', TaxCloud Enterprise checkbox, Cash Basis checkbox, Rounding Method options), 'Currencies' (Main Currency set to USD, Multi-Currencies checkbox), and 'Fiscal Periods'.

Under **Currencies**, you will find **Multi-Currencies**. By checking this, you will get the option to select the accounts to which the differences between exchange rates are posted. When setting up a full production system, you will want to assign the appropriate accounts, as discussed in [Chapter 6, Configuring Accounting - Finance Options](#). However, we can post the **Exchange Difference** to the default account as an appropriate typed income or expense account for demonstration purposes.

After clicking on **Multi-Currencies**, the screen will refresh and give you the option to **Activate Other Currencies**. Clicking this option will bring up all the currencies that are available in your Odoo installation:

The screenshot shows the Odoo web interface for managing currencies. The title bar says "Currencies - Odoo". The URL is "localhost:8069/web?debug=#action=65&active_id=9&model=res.currency&view_type=list". The top navigation bar includes "Settings", "Dashboard", "Users & Companies", "Translations", "General Settings", "Technical", and "Administrator (SILKWORM-DEV)". Below the navigation is a search bar and filter options. The main content area is titled "Settings / Currencies" and contains a table with columns: "Currency", "Symbol", "Date", "Current Rate", and "Active". The table lists numerous currencies, each with a checkbox in the first column. Most currencies have their "Active" status set to 1.000000, indicated by a blue circle. A few currencies like AED, AFN, ALL, AMD, ANG, AOA, ARS, AUD, AWG, AZN, BAM, BBD, BDT, BGN, BHD, BIF, BMD, BND, BOB, BRL, BSD, BTN, BWP, BYN, and BYR have their "Active" status set to 0.000000, indicated by a grey circle.

Currency	Symbol	Date	Current Rate	Active
EUR	€		1.000000	●
USD	\$		1.000000	●
AED	₼		1.000000	○
AFN	Afs		1.000000	○
ALL	L		1.000000	○
AMD	դր.		1.000000	○
ANG	f		1.000000	○
AOA	Kz		1.000000	○
ARS	\$		1.000000	○
AUD	\$		1.000000	○
AWG	Afl.		1.000000	○
AZN	m		1.000000	○
BAM	KM		1.000000	○
BBD	Bds\$		1.000000	○
BDT	৳		1.000000	○
BGN	лв		1.000000	○
BHD	BD		1.000000	○
BIF	FBu		1.000000	○
BMD	BD\$		1.000000	○
BND	\$		1.000000	○
BOB	Bs.		1.000000	○
BRL	R\$		1.000000	○
BSD	B\$		1.000000	○
BTN	Nu.		1.000000	○
BWP	P		1.000000	○
BYN	Br		1.000000	○
BYR	BR		1.000000	○

When you turn on **Multi-currency**, Odoo specifies both the **USD** and **EUR** currencies as active. You can activate any other currency by editing the appropriate record and checking the active option.

Let's go ahead and activate the currency for the Mexican Peso by finding the **MXN** currency record and setting it to **Active**, as shown in the following screenshot:

The screenshot shows the Odoo web interface for managing currencies. The top navigation bar includes links for Dashboard, Users & Companies, Translations, General Settings, Technical, and a user icon for Administrator (SILKWORM-DEV). The main title is "Settings / Currencies / MXN". Below the title, there are "Edit" and "Create" buttons and a "Action" dropdown menu. The main content area displays the following information:

Currency	MXN	Currency Unit	Pesos
Current Rate	1.000000	Currency Subunit	Centavos
Price Accuracy		Display	
Rounding Factor	0.010000	Symbol	\$
Decimal Places	2	Symbol Position	Before Amount

A small "Inactive" button is visible in the top right corner of the main content area.

You will notice that this screen also contains the option to specify the rounding factor that will be used for converting the **MXN** currency into the base Odoo currency. In this installation, this is the **US** currency.

You won't notice the changes from **Multi-currency** until we look at how some of our documents now appear in Odoo.



Like when we're configuring other Odoo options, it is a good idea to *Shift*-refresh your browser to force Odoo to load any new menus or settings resulting from your changes.

Purchasing in a different currency

Create a new purchase order and observe the new **Currency** selection that's available at the top of the form, as shown in the following screenshot:

The screenshot shows the Odoo web interface for creating a new purchase order. At the top, there's a header bar with tabs for Purchase, Purchase, Control, Reporting, Configuration, and Administrator (SILKWORM-DEV). Below the header, the URL is localhost:8069/web?debug=#id=&action=336&model=purchase.order&view_type=. The main content area is titled "Requests for Quotation / New". It has buttons for Save, Discard, Send by Email, Print RFQ, Confirm Order, and Cancel. A progress bar at the top right indicates the process: RFQ → RFQ Sent → Purchase Order. The main form is titled "New" and contains fields for Vendor (T-Shirt Supply Co.), Vendor Reference (empty), and Currency (MXN). Below the form is a table with columns: Product, Description, Scheduled Date, Quantity, Product Unit of Measure, Unit Price, Taxes, and Subtotal. A note says "Add a line" and there are three empty lines for entering products. At the bottom, there's a text area for "Define your terms and conditions ..." and a summary table with Untaxed Amount: 0.00, Taxes: 0.00, and Total: 0.00. At the very bottom, there are buttons for Send message, Log note, Schedule activity, Follow, and a notification count of 0.

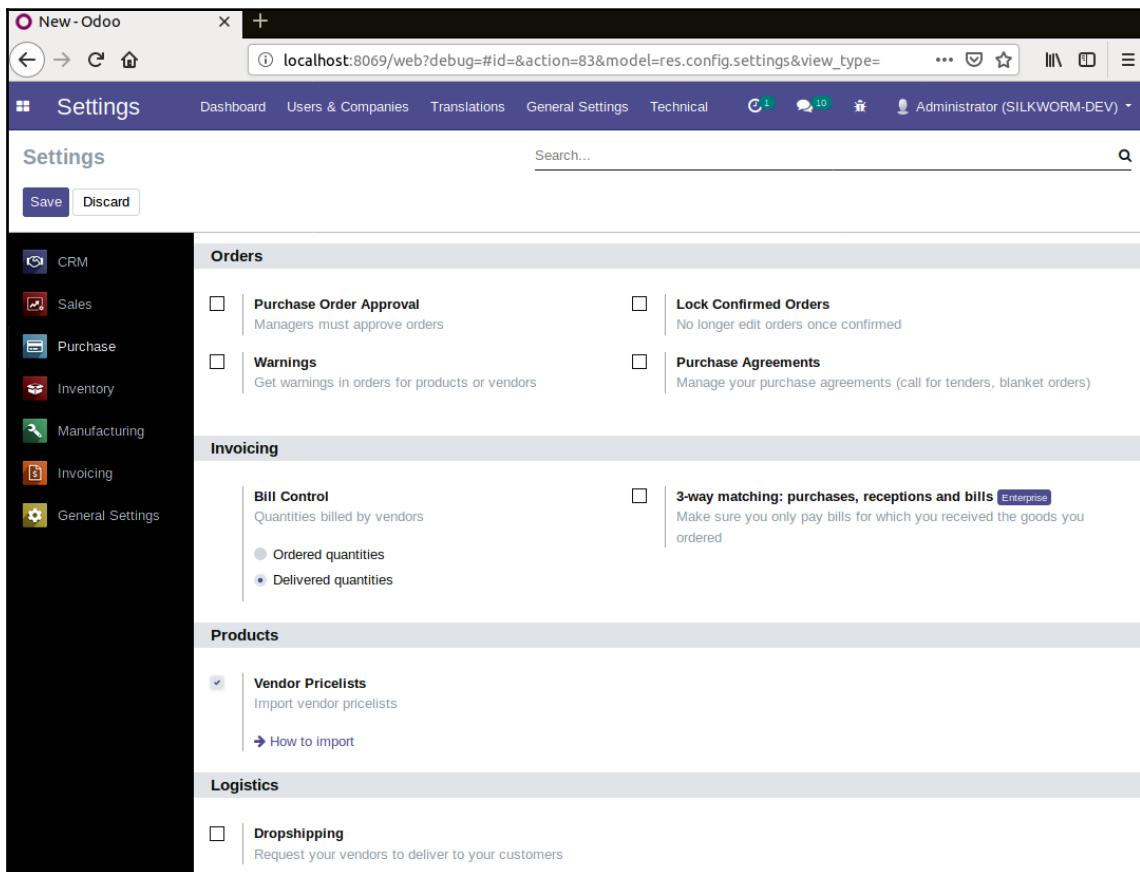
Once you have chosen a **Currency**, you will see the symbol change at the bottom of the purchase order to show that you are now purchasing in the new currency.

However, if you then try to add a product to your purchase order, you will quickly discover that the unit cost does not auto-populate from the cost we specified in the product file. Since we are now using multiple currencies, we either need to provide more detailed pricing information, or we must enter the amount on each purchase order we create.

Managing supplier price lists

Now that we know we are going to have suppliers from whom we need to purchase in a different currency, let's see how we can set up a price list for an alternative currency so that when we order products, we don't have to re-enter our costs.

First, we must turn on multiple vendor price lists. Click on the **Purchases** menu and then click **Settings** within the **Configuration** section. There, you can check the **Allow using and importing vendor pricelists** option under the **Vendor Price** section and click **Apply**:



The screenshot shows the Odoo Settings page under the Configuration menu. On the left sidebar, there are icons for CRM, Sales, Purchase, Inventory, Manufacturing, Invoicing, and General Settings. The main content area has tabs for Orders, Invoicing, Products, and Logistics. Under the Orders tab, there are sections for Purchase Order Approval, Warnings, Lock Confirmed Orders, and Purchase Agreements. Under the Invoicing tab, there are sections for Bill Control and 3-way matching. Under the Products tab, there is a section for Vendor Pricelists, which includes a link to 'How to import'. Under the Logistics tab, there is a section for Dropshipping. A sidebar on the left contains a 'Save' button and a 'Discard' button. The URL in the browser bar is `localhost:8069/web?debug=#id=&action=83&model=res.config.settings&view_type=`.

After we have turned on the option to manage price lists per supplier, we can go to the vendor and set the **Supplier Currency** for the vendor under the **Sales & Purchases** tab:

The screenshot shows the Odoo Partner form for 'T-Shirt Supply Co.' (id: 11). The top navigation bar includes links for Purchase, Purchase Control, Reporting, Configuration, and Administrator (SILKWORM-DEV). Below the header, there are tabs for Vendors / T-Shirt Supply Co., Save, Discard, and a status indicator (1 / 1).

The main content area displays the vendor's details. On the left, there is a logo icon and a radio button for 'Company'. The vendor name 'T-Shirt Supply Co.' is prominently displayed. To the right, there are several status indicators: 0 Meetings, 2 Purchases, 1 Vendor Bills, 0 Analytic Acc..., 0 Credit card(s), and Active.

Below the status indicators, there are sections for Address, Phone, Mobile, Email, Website, Language, and Tags. The 'Address' section contains fields for Street, City, State, Zip, and Tax ID. The 'Phone' section contains fields for Main Phone and Mobile. The 'Language' section shows English selected. The 'Tags' section has a dropdown menu.

At the bottom of the form, there are tabs for Contacts & Addresses, Internal Notes, Sales & Purchases (which is currently selected), and Invoicing. The 'Sales' section includes fields for Is a Customer (checkbox), Salesperson (dropdown), Bounce (text input), Payment Terms (dropdown), and Pricelist (dropdown). The 'Purchase' section includes fields for Is a Vendor (checkbox), Payment Terms (dropdown), and Supplier Currency (dropdown set to MXN). The 'Misc' section includes fields for Internal Reference and Barcode. The 'Fiscal Information' section includes a Fiscal Position dropdown.

Once you save the vendor record and go to create a new purchase order, the currency will automatically default to the new supplier currency we specified.

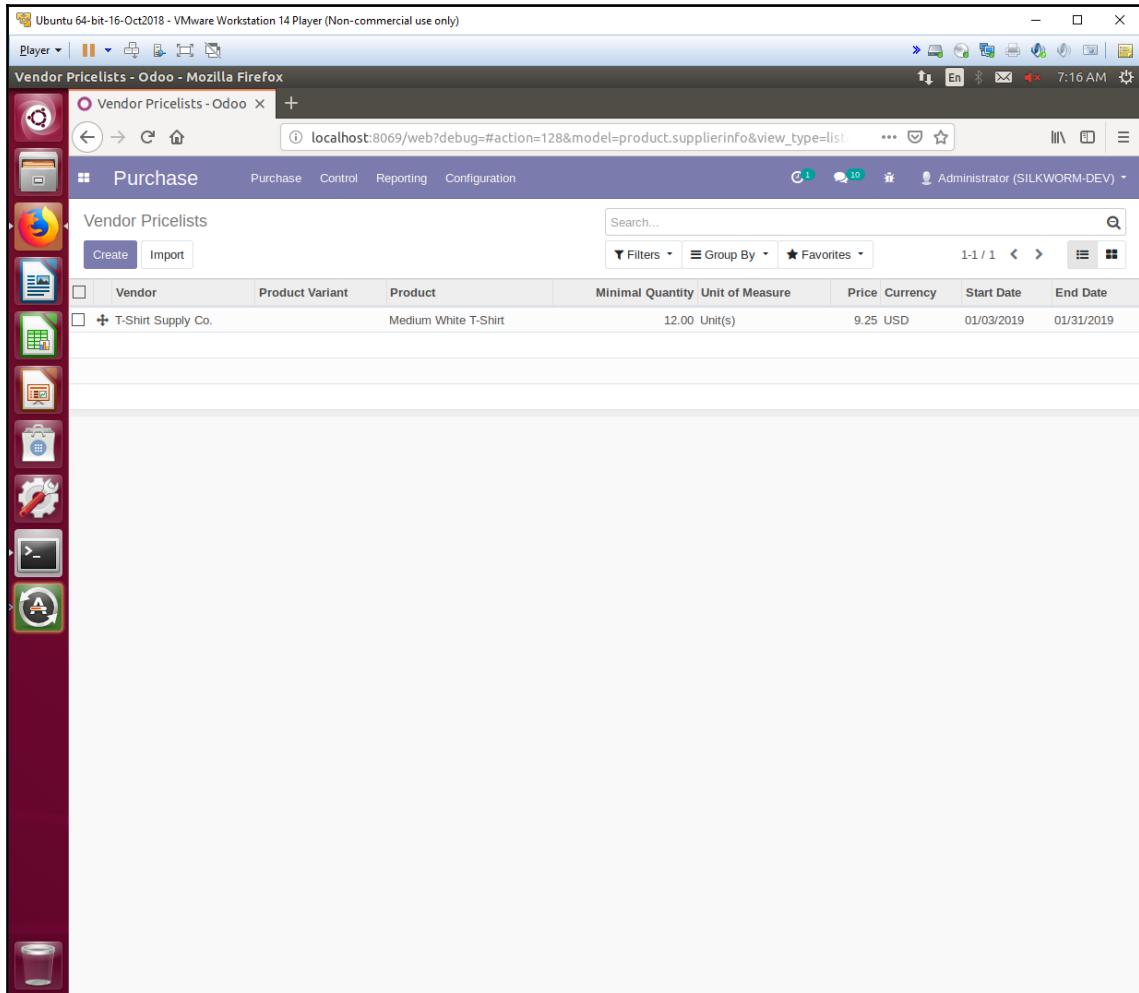
Let's see what happens when we add the line item for the medium white T-shirt:

The screenshot shows the Odoo web interface for creating a purchase order (PO00002). The vendor is set to 'T-Shirt Supply Co.' with an order date of '02/01/2019 07:06:45'. A single line item is added for a 'MED WHT SHIRT' (product code [MWT-20]), quantity 12.000, unit price 0.00, and taxes applied at 15.00%. The total subtotal is 0.00. The bottom right corner indicates the user is 'Following' the document.

Product	Description	Scheduled Date	Quantity	Product Unit of Measure	Unit Price	Taxes	Subtotal
[MWT-20]	[MWT-20] MED WHT SHIRT	02/01/2019 07:13:50	12.000	Unit(s)	0.00	Tax 15.00%	0.00

If you look carefully at the preceding screenshot, you will notice that the **Unit Price** for our medium white T-shirt is still set to **0.00**. *What's happened?* While we configured the vendor to use the **MXN** currency, we only specified a cost for **US** currency. Let's see how we can fix that.

Under the **Purchase** menu, select **Vendor Pricelists**, as follows:



Let's make it so that this shirt is now priced in Peso (MXN):

The screenshot shows the Odoo web interface for the 'Purchase' module. The title bar says 'T-Shirt Supply Co. - Odoo'. The top navigation bar includes 'Purchase', 'Control', 'Reporting', and 'Configuration'. On the right, there are icons for 'Administrator (SILKWORM-DEV)', 'EV', and a search bar. Below the navigation is a breadcrumb trail: 'Vendor Pricelists / T-Shirt Supply Co.'. A toolbar with 'Save' and 'Discard' buttons is at the top left, and a page number '1 / 1' is at the top right.

The main form is divided into two sections: 'Vendor' on the left and 'Price List' on the right.

Vendor Section:

- Product Variant: dropdown menu
- Vendor: dropdown menu set to 'T-Shirt Supply Co.'
- Vendor Product Name: input field set to 'MED WHT SHIRT'
- Vendor Product Code: input field set to 'MWT-20'
- Delivery Lead Time: input field set to '4 days'

Price List Section:

- Product: dropdown menu set to 'Medium White T-Shirt'
- Minimal Quantity: dropdown menu set to '12.00 Unit(s)'
- Price: dropdown menu set to '70.25'
- Currency: dropdown menu set to 'MXN'
- Validity: date range from '01/03/2019' to '02/28/2019'

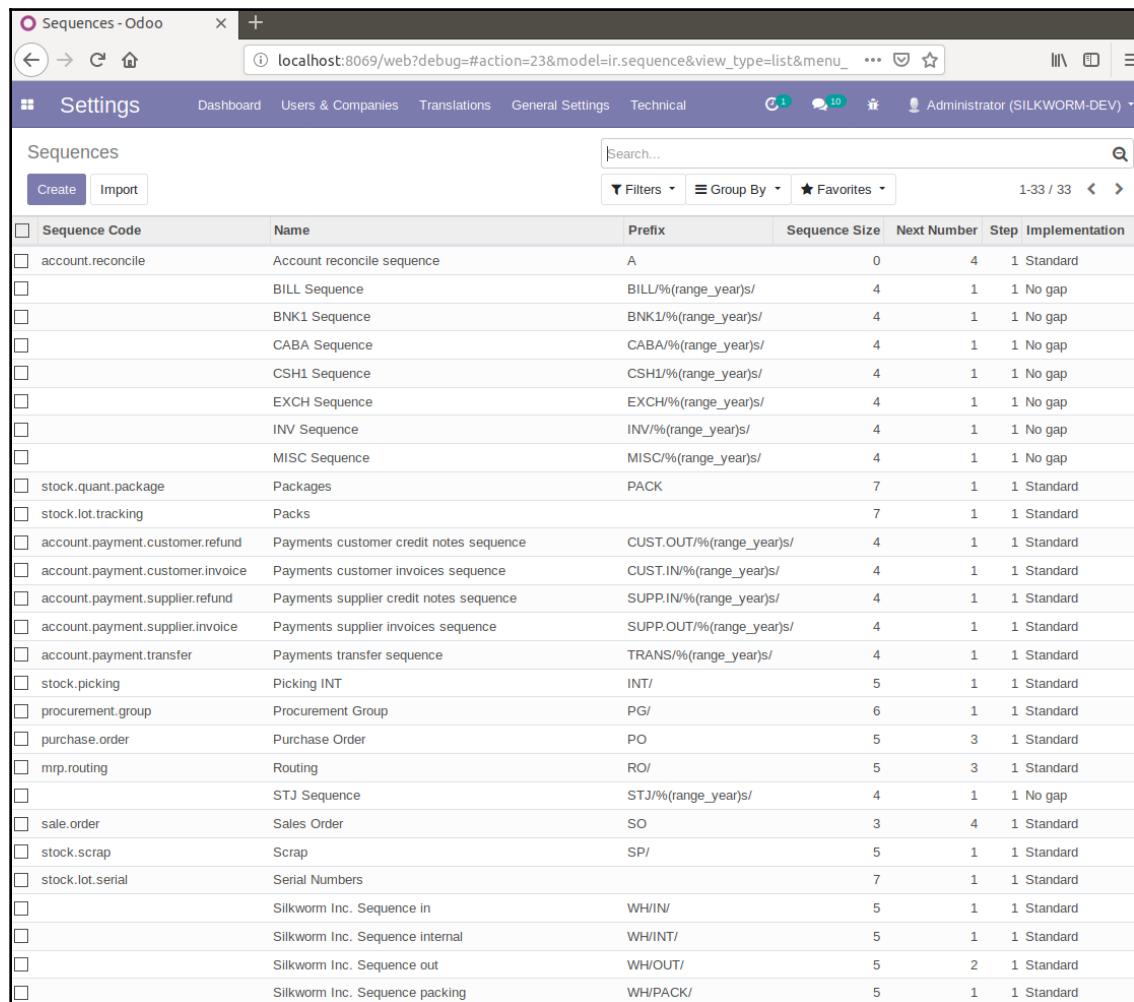
Once you **Save** and create a new purchase order, the product will then be picked up and priced correctly in Pesos.

As you can see, price lists are flexible, but you must be careful to properly configure and test your configuration. It's very easy to end up pulling the wrong currency into a document.

Managing sequences in Odoo

When you are setting up a system for your business, there's a good chance that the default naming conventions of documents and number sequences that are used by Odoo are not ideal. A simple example is that you're unlikely to want your invoices starting at 00001 if you've already produced thousands of invoices. In this case, you would want the number to start where the old system left off.

Additionally, sequences in Odoo don't just manage the numbering of your documents. They also manage how the document name looks inside of Odoo. To see the current sequences defined by Odoo, go to the **Settings | Technical** menu and choose **Sequences** under **Sequences & Identifiers**. In Odoo 12, this is only accessible through Developer Mode. Here you will find a list of all documents within the system that are auto-assigned document numbers. You can prime each sequence with where to start numerically (**Next Number**). You can also set what **Prefix** you prefer to use to denote the type of document, how much space to pad on the left of the document number (**Sequence Size**), how many digits to increment (**Step**), and whether the system will allow gaps between document numbers (**Implementation**):



The screenshot shows the Odoo web interface with the URL `localhost:8069/web?debug=#action=23&model=ir.sequence&view_type=list&menu_...`. The page title is "Sequences - Odoo". The top navigation bar includes "Settings", "Dashboard", "Users & Companies", "Translations", "General Settings", "Technical", and "Administrator (SILKWORM-DEV)". The main content area is titled "Sequences" and displays a list of sequences with columns: Sequence Code, Name, Prefix, Sequence Size, Next Number, Step, and Implementation. The table lists various sequences such as "account.reconcile", "BILL Sequence", "BNK1 Sequence", etc., with their respective settings.

<input type="checkbox"/> Sequence Code	Name	Prefix	Sequence Size	Next Number	Step	Implementation
<input type="checkbox"/> account.reconcile	Account reconcile sequence	A	0	4	1	Standard
<input type="checkbox"/>	BILL Sequence	BILL%(range_year)s/	4	1	1	No gap
<input type="checkbox"/>	BNK1 Sequence	BNK1%(range_year)s/	4	1	1	No gap
<input type="checkbox"/>	CABA Sequence	CABA%(range_year)s/	4	1	1	No gap
<input type="checkbox"/>	CSH1 Sequence	CSH1%(range_year)s/	4	1	1	No gap
<input type="checkbox"/>	EXCH Sequence	EXCH%(range_year)s/	4	1	1	No gap
<input type="checkbox"/>	INV Sequence	INV%(range_year)s/	4	1	1	No gap
<input type="checkbox"/>	MISC Sequence	MISC%(range_year)s/	4	1	1	No gap
<input type="checkbox"/> stock.quant.package	Packages	PACK	7	1	1	Standard
<input type="checkbox"/> stock.lot.tracking	Packs		7	1	1	Standard
<input type="checkbox"/> account.payment.customer.refund	Payments customer credit notes sequence	CUST.OUT%(range_years)s/	4	1	1	Standard
<input type="checkbox"/> account.payment.customer.invoice	Payments customer invoices sequence	CUST.IN%(range_years)s/	4	1	1	Standard
<input type="checkbox"/> account.payment.supplier.refund	Payments supplier credit notes sequence	SUPP.IN%(range_years)s/	4	1	1	Standard
<input type="checkbox"/> account.payment.supplier.invoice	Payments supplier invoices sequence	SUPP.OUT%(range_years)s/	4	1	1	Standard
<input type="checkbox"/> account.payment.transfer	Payments transfer sequence	TRANS%(range_years)s/	4	1	1	Standard
<input type="checkbox"/> stock.picking	Picking INT	INT/	5	1	1	Standard
<input type="checkbox"/> procurement.group	Procurement Group	PG/	6	1	1	Standard
<input type="checkbox"/> purchase.order	Purchase Order	PO	5	3	1	Standard
<input type="checkbox"/> mrp.routing	Routing	RO/	5	3	1	Standard
<input type="checkbox"/>	STJ Sequence	STJ%(range_year)s/	4	1	1	No gap
<input type="checkbox"/> sale.order	Sales Order	SO	3	4	1	Standard
<input type="checkbox"/> stock.scrap	Scrap	SP/	5	1	1	Standard
<input type="checkbox"/> stock.lot.serial	Serial Numbers		7	1	1	Standard
<input type="checkbox"/>	Silkworm Inc. Sequence in	WH/IN/	5	1	1	Standard
<input type="checkbox"/>	Silkworm Inc. Sequence internal	WH/INT/	5	1	1	Standard
<input type="checkbox"/>	Silkworm Inc. Sequence out	WH/OUT/	5	2	1	Standard
<input type="checkbox"/>	Silkworm Inc. Sequence packing	WH/PACK/	5	1	1	Standard

Here, we have scrolled down so that you can see the **Sales Order** sequence. Click on the **Sales Order** sequence to bring up the details, as shown in the following screenshot:

The screenshot shows the Odoo web interface for managing sequences. The top navigation bar includes links for Sales Order - Odoo, Settings, Dashboard, Users & Companies, Translations, General Settings, Technical, and Administrator (SILKWORM-DEV). The main content area is titled "Sequences / Sales Order". A sub-menu bar at the top of the form has "Save" and "Discard" buttons, along with a page number "21 / 33".

The configuration fields for the "Sales Order" sequence are:

- Name:** Sales Order
- Implementation:** Standard
- Sequence Code:** sale.order
- Active:** checked
- Prefix:** SILK-SO
- Sequence Size:** 3
- Suffix:** (empty)
- Step:** 1
- Use subsequences per date_range:** unchecked
- Next Number:** 5200

Legend (for prefix, suffix)

Current Year with Century: %(year)s	Day of the Year: %(doy)s	Hour 00->24: %(h24)s
Current Year without Century: %(y)s	Week of the Year: %(woy)s	Hour 00->12: %(h12)s
Month: %(month)s	Day of the Week (0:Monday): %(weekday)s	Minute: %(min)s
Day: %(day)s		Second: %(sec)s

Simply using **SO** is generic and could potentially match a document identifier from another company.

So, for this example, we have changed the **Prefix** of the **Sales Order** in our Odoo installation so that it begins with **SILK-SO**. In a business situation, it's common that you'll want to prefix your documents with a notation that identifies that document as being for your company. Also note that we have bumped up the **Next Number** to 5200:

The screenshot shows the Odoo web interface for a Sales Order. The title bar indicates the window is titled 'SILK-SO5200 - Odoo'. The URL in the browser is 'localhost:8069/web?debug=#id=4&action=257&model=sale.order&view_type=form'. The top navigation bar includes 'Sales', 'Orders', 'To Invoice', 'Products', 'Reporting', and 'Configuration'. On the right, there are notifications for 1 message and 10 unread emails, and the user is logged in as 'Administrator (SILKWORM-DEV)'. Below the navigation, the page title is 'Quotations / SILK-SO5200'. There are buttons for 'Edit' and 'Create', and links for 'Print', 'Action', 'Send by Email', 'Print', 'Confirm', 'Preview', and 'Cancel'. A breadcrumb navigation shows 'Quotation' > 'Quotation Sent' > 'Sales Order'. The main content area displays a quotation for 'SILK-SO5200'. It shows the customer information: 'Creative Hair Designs, 440 South Hampton, Carbondale IL 62901, United States'. It also shows the validity period, pricelist ('Public Pricelist (USD)'), and payment terms. The order lines section lists a single item: 'Medium White T-Shirt' ordered at 1.000 Unit(s) for \$16.50, with a tax of 15.00%, resulting in a subtotal of \$16.50. The total amount is \$18.98. At the bottom, there are buttons for 'Send message', 'Log note', and 'Schedule activity', and a status indicator showing 'Following' with 3 users.

After you save the changes to the sequence, any new documents will now use the new sequence definition, as shown in the preceding screenshot.

Multiple companies in Odoo

Odoo can manage multiple companies within the same database. This feature allows you to consolidate some of your system administration and manage more complex operations. As a rule, multiple-company configuration is an advanced topic. You should be very comfortable working with single-company configurations before you begin looking into multiple-company configurations. When managing operations, Odoo's warehouse management and analytic accounting abilities are often preferable to configuring multiple companies.

A good general rule is that if a company is not a separate legal entity, then it shouldn't be set up as part of multiple companies in Odoo. However, every business requirement is different, and the ability to use multiple companies in Odoo may allow you to easily implement a solution that otherwise may have been difficult.

Setting up a second company in Odoo

Now, it's time to create a second company. With Odoo, you can have multiple companies that are all independent of each other, or you can have child companies in which you can link a chart of accounts and other operations to a parent company. For example, we have created a new company named Euro Shirts and have set Silkworm Inc. as the parent company.

Go to **Settings**. Then, under **Users & Companies**, choose **Companies** and click **Create** to create a new company.

In this example, we are choosing **France** as the country:

The screenshot shows the Odoo web interface for managing company settings. The title bar indicates the window is for 'Euro Shirts - Odoo'. The URL in the address bar is 'localhost:8069/web?debug=#id=2&action=48&model=res.company&view_type=form'. The top navigation bar includes links for Settings, Dashboard, Users & Companies, Translations, General Settings, Technical, and a notifications icon showing 10 messages. The user is logged in as 'Administrator (SILKWORM-DEV)'. The main content area is titled 'Companies / Euro Shirts' and shows a preview image of a camera with a plus sign. Below this, the company name 'Euro Shirts' is displayed in a large blue header. A 'General Information' section contains fields for Partner (labeled 'Euro Shirts'), Address ('4300 Place Bellecour', 'Street 2...', 'Lyon', 'France'), Website (text input), Phone (text input), Email (text input), Tax ID (text input), Company Registry (text input), Default Incoterm (dropdown menu), Currency ('EUR' selected), and Nomenclature (dropdown menu). The 'Country' field in the address section is explicitly set to 'France'.

In the following screenshot, we have assigned the user a second company:

The screenshot shows the Odoo web interface for managing user settings. The URL in the browser is `localhost:8069/web?debug=#id=6&action=68&model=res.users&view_type=form&`. The top navigation bar includes links for Dashboard, Users & Companies, Translations, General Settings, Technical, and a message center with 1 notification. The current user is listed as Administrator (SILKWORM-DEV).

The main content area is titled "Users / Terry Zeigler". It displays the user's details: Name (Terry Zeigler), Email Address (terryzeigler.exampleemail.com), and Status (Active). Below this, the "Related Partner" field is set to Terry Zeigler.

The "Multi Companies" section shows the user is allowed to access Silkworm Inc. and Euro Shirts. The "Current Company" dropdown is also set to Silkworm Inc.

The "User Type" section indicates the user is an Internal User.

The "Application Accesses" section lists various Odoo modules and their assigned roles: Sales (Manager), Inventory (empty), Manufacturing (empty), Accounting & Finance (Billing Manager), and Purchases (empty).

Once you have added in the second company, you may need to *Shift-refresh* your browser. At that point, you can choose which company you are working with via a drop-down box on the right-hand side of the window:

The screenshot shows the Odoo web interface for managing users. The top navigation bar includes links for Settings, Dashboard, Users & Companies, Translations, General Settings, and Technical. A dropdown menu titled 'Euro Shirts' is open, showing two options: 'Euro Shirts' (selected) and 'Silkworm Inc.'. The main content area displays a user profile for 'Terry Zeigler' (terryzeigler@examplemail.com). The 'Multi Companies' section lists 'Allowed Companies' as 'Euro Shirts' and 'Silkworm Inc.', with 'Silkworm Inc.' currently selected as the 'Current Company'. Under 'User Type', 'User types' are set to 'Internal User'. In the 'Application Accesses' section, roles are assigned to various departments: Sales (Manager), Inventory, Manufacturing, Accounting & Finance (Billing Manager), Purchases, and Administration. The 'Technical Settings' section contains a note about warnings for partners related to Account and Stock.

Now that we have defined a second company, we need to set up the **Fiscal Localization**.

Setting up the chart of accounts for your second company

When we created our database in Chapter 2, *Installing Your First Application*, Odoo installed the **United States Fiscal Localization** based on the country we chose in the setup. Go to the **Invoicing** menu and choose **Settings** under **Configuration**. At the top left, you will see the option to choose the **Fiscal Localization** for Euro Shirts:

The screenshot shows the Odoo web interface with the title bar "localhost:8069/web?debug=#id=&action=83&model=res.config.settings&view_type=kanban". The main navigation bar includes "Settings", "Dashboard", "Users & Companies", "Translations", "General Settings", and "Technical". On the left, a sidebar lists modules: CRM, Sales, Purchase, Inventory, Manufacturing, Invoicing, and General Settings. The "Invoicing" module is currently selected. The main content area is titled "Settings" and contains three sections: "Fiscal Localization", "Setup", and "Taxes".

- Fiscal Localization:** A section titled "Fiscal Localization" with a sub-section "Fiscal Localization". It includes a "Package" dropdown menu and a link "Install More Packages".
- Setup:** A section titled "Bank Reconciliation Threshold" which describes payments not matched with bank statements before a certain date.
- Taxes:** A section with two main parts:
 - Default Taxes:** Describes default taxes applied to local transactions. It includes dropdown menus for "Sales Tax" and "Purchase Tax".
 - Rounding Method:** Describes how total tax amounts are computed. It includes radio buttons for "Round per Line" (selected) and "Round Globally".

Since we haven't set the **Fiscal Localization** yet, the **Package** dropdown has no current selection. Click **Install More Packages** to bring up a list of available **Chart Templates**:

The screenshot shows the Odoo Settings / Chart Templates page. At the top, there is a search bar with filters, group by, and favorite options. Below the search bar is a grid of 24 chart template modules, each represented by a small icon, the name of the template, its localization code, and two buttons: 'Install' and 'Module Info'. The modules listed include:

- U.A.E. - Accounting (lion_ae)
- Argentina - Accounting (lion_ar)
- Austria - Accounting (lion_at)
- Australian - Accounting (lion_au)
- Belgium - Accounting (lion_be)
- Bolivia - Accounting (lion_bo)
- Brazilian - Accounting (lion_br)
- Canada - Accounting (lion_ca)
- Switzerland - Accounting (lion_ch)
- Chile - Accounting (lion_cl)
- China - Accounting (lion_cn)
- China - City Data (lion_cn_city)
- China - Small Business CoA (lion_cn_sb)
- Colombian - Accounting (lion_co)
- Costa Rica - Accounting (lion_cr)
- Germany - Accounting (lion_de)
- Germany SKR03 - Accounting (lion_de_sk03)
- Germany SKR04 - Accounting (lion_de_sk04)
- Dominican Republic - Accounting (lion_do)
- Ecuador - Accounting (lion_ec)
- Spain - Accounting (PGCE 2012) (lion_es)
- Ethiopia - Accounting (lion_et)
- EU Mini One Stop Shop (MOS) (lion_eu)
- France - Accounting (lion_fr)
- Generic - Accounting (lion_g)
- Greece - Accounting (lion_gr)
- Guatemala - Accounting (lion_gt)
- Hong Kong - Accounting (lion_hk)
- Honduras - Accounting (lion_hn)
- Croatia - Accounting (RRIF 2012) (lion_hr)
- Hungarian - Accounting (lion_hu)

Now, you can search for the **Chart Templates** that you require for your second company. In this case, we will install the **Chart Templates for France**:

The screenshot shows the Odoo Settings / Chart Templates page with a search bar containing 'france'. The results show only one module: 'France - Accounting (lion_fr)'. This module has an 'Installed' status and two buttons: 'Install' and 'Module Info'.

It will take a few moments for Odoo to complete the configuration. Now, when you look at the **Settings** for **Euro Shirts**, you will see that the new template has been applied:

The screenshot shows the Odoo web interface for the 'Settings' module. The left sidebar lists various modules: CRM, Sales, Purchase, Inventory, Manufacturing, Invoicing, and General Settings. The main content area is titled 'Fiscal Localization' under the 'Setup' tab. It includes sections for 'Fiscal Localization' (describing taxes, fiscal positions, chart of accounts & legal statements for your country), 'Bank Reconciliation Threshold' (payments not matched with a bank statement before a certain date), and 'Taxes'. Under 'Taxes', there are sections for 'Default Taxes' (Sales Tax: TVA collectée (vente) 20.0%, Purchase Tax: TVA déductible (achat) 20.0%), 'Rounding Method' (Round per Line or Round Globally selected), 'EU Digital Goods VAT' (Apply right VAT rates for digital products sold in EU), and 'Cash Basis' (Tax Cash Basis Journal: Cash Basis Tax Journal (EUR)). At the top of the page, the URL is localhost:8069/web?debug=true#id=83&action=83&model=res.config.settings&view_type=form&menu_... and the top bar shows the user is 'Administrator (SILKWORM-DEV)'.

We have now completed several of the basic steps that are required to implement a multi-company solution in Odoo.

Implementing a multi-company solution

As we stated previously, setting up a multi-company system is complex. While the system will work much the same as it did previously, it is important that you understand how a multi-company system impacts every operation within the system. Customers, users, suppliers, and the chart of accounts all tie in to multiple-company operations. This chapter has provided the very basics of how to configure a multi-company setup, but final configurations will take a great deal of planning and fine-tuning if you want a truly successful installation.

Summary

In this chapter, we examined some of the things you should consider when administering an Odoo installation, such as planning your server configuration and establishing good practices for ensuring business continuity in the case of failure. We discovered how to back up and restore databases, as well as how to manage user access and group permissions.

Later in this chapter, we looked at internationalization and configured Odoo to handle multiple languages and international currency. We learned how to change Odoo sequences so that our documents will use formats and numbering systems that work for our business requirements. Finally, we took a brief look at how to set up a multi-company configuration in Odoo.

In the next chapter, we will look at human resources and how we can apply Odoo to make recruiting, interviewing, hiring, and managing employees easier.

8

Implementing the Human Resources Application

Over the past few decades, companies have had increasing demands placed upon them to keep track of employee-related information. Odoo has a variety of modules that can help your company to organize information involving your employees. Some of these applications, such as timesheets and attendance, can become critical processes that help a company to contain costs. In this chapter, we will look at how you can integrate **Human Resource (HR)** applications.

In this chapter, we will cover the following:

- The functions of the Employee Directory
- Creating timesheets to track time spent on projects
- How to use leave management to manage employee vacations and time off
- Recruiting new employees to fill job vacancies

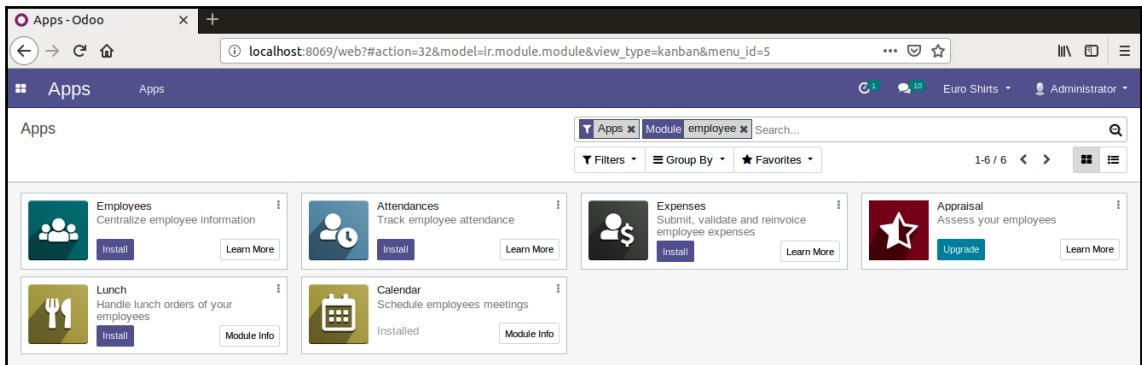
Modular approach to HR applications

Like the rest of Odoo, the HR applications allow you to implement the functionality you need today, and then later add additional modules. This approach makes it much easier to start using Odoo right away to solve specific company needs. The best way to be successful with implementing a system is to plan ahead and implement in stages. Once you are successful at putting one application in place, then you can move on to putting additional applications in place.

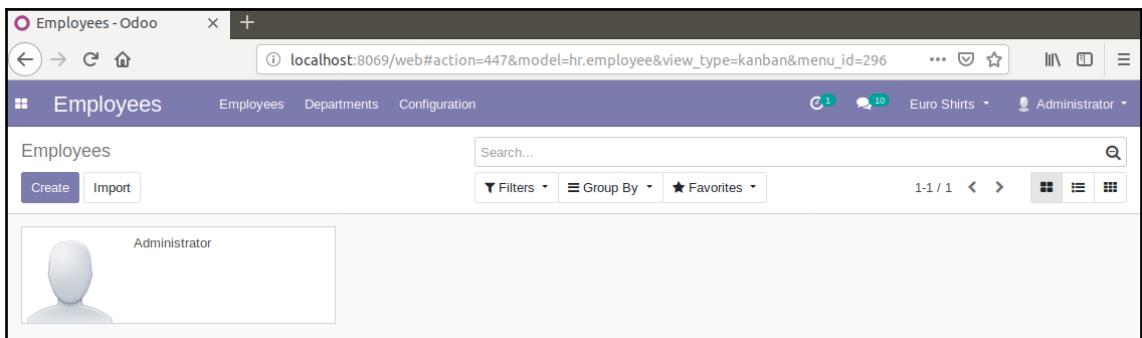
Installing the Employee Directory

When you install the base Odoo applications, you get the ability to manage system users, but you will notice that there are no menu options for entering and managing employees. To begin working with HR applications, you will need to install the base **Employee Directory** application.

Go to the **Settings** menu and install the **Employee Directory** application using the same process as the previous Odoo applications:



After you have installed the **Employee Directory** application, you will see a new menu at the top called **Employees**. Clicking on the **Menu** will take you to a dashboard that displays the **Employees** in a kanban view as follows:



What you may notice first is that, for better or for worse, Odoo does not consider users employees. The only employee that is in the current Odoo installation is the **Administrator**. On top of that, there is no built-in mechanism to automatically turn your users into employees.

Creating a new employee

Let's go ahead and see how we can add a new employee to our system.

Click the **Create** button to bring up the form for you to begin entering a new employee into Odoo:

The screenshot shows the Odoo web interface for creating a new employee. The browser address bar indicates the URL is `localhost:8069/web#id=&action=447&model=hr.employee&view_type=form&menu_id=296`. The page title is "Employees / New". The main form has a purple header bar with the name "Tina Robbins". Below the header, there are tabs for "Work Information", "Private Information", and "HR Settings". The "Contact Information" section includes fields for Work Address (Silkworm Inc.), Work Location, Work Email, Work Mobile, and Work Phone. The "Position" section includes fields for Department (checkbox), Job Position, Job Title, Manager, Coach, Working Hours (Standard 40 hours/week), and Timezone (UTC). A large text area at the bottom is labeled "Other Information ...". At the bottom of the form, there are buttons for "Send message", "Log note", "Schedule activity", "Follow" (with 0 notifications), and a date selector set to "Today".

The only required field in the employee form is the name. All of the other fields are optional. Odoo will default the working address to the company address. While most fields are self-explanatory, we will go over several of the more important fields to take into consideration.

Department

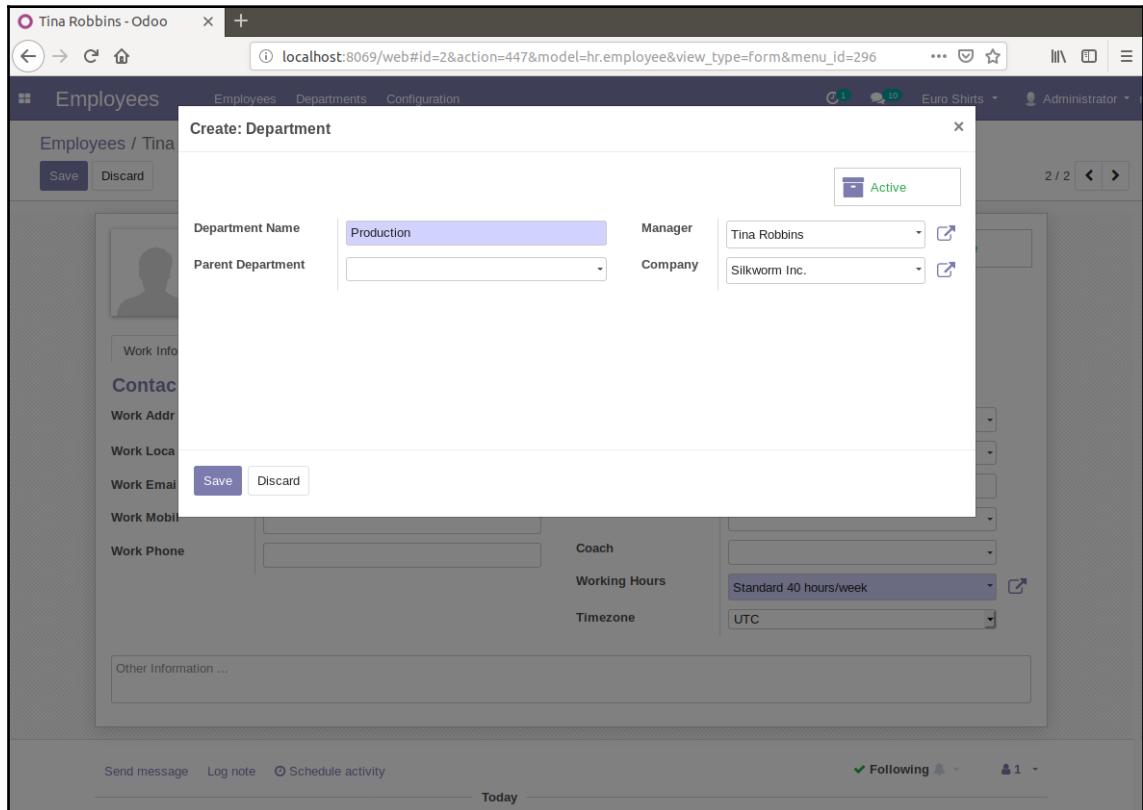
The employee department is a common way for a company to organize employees. In our example, we are going to create a production department and assign Tina Robbins to that department. After you have saved the record for Tina Robbins, edit it once again and use the **Create and Edit...** option in the **Department** field to add a department:

The screenshot shows the Odoo web interface for creating a new employee. The URL is `localhost:8069/web#id=&action=447&model=hr.employee&view_type=form&menu_id=296`. The top navigation bar includes links for Employees, Departments, and Configuration, along with a notification for 10 messages and a user named Administrator.

The main form is titled "Employees / New". It has fields for Name (Tina Robbins), Department (Production), and other contact and position details. The "Position" section includes fields for Department (with a dropdown menu showing "Create 'Production'" and "Create and Edit..."), Job Position, Job Title, Manager, Coach, Working Hours (set to "Standard 40 hours/week"), and Timezone (set to "UTC").

At the bottom of the form, there are buttons for "Send message", "Log note", "Schedule activity", "Follow" (with 0 notifications), and a date field set to "Today".

In this screen, we have set the **Department Name** to Production and set the **Manager** of this department to **Tina Robbins**. Please be aware that you must save the record first before you can assign **Tina Robbins** as the manager:



Also, you will notice the **Parent Department** field. This field allows you to create a hierarchical structure of departments for your company. Typically, you will wish to look at the organization chart of a company and take some time preparing the company department structure.

Job Title

The **Job Title** field allows you to manage job titles for employees inside of Odoo:

The screenshot shows the Odoo interface for creating a new job position. The main window is titled "Create: Job Position". The job title "Production Manager" is highlighted in a blue box. The "Recruitment" section includes fields for Company (Silkworm Inc.), Department (Production), and Expected New Employees (1). The "Job Description" field contains the text: "This position supervises the production department and reports to the Director of Operations." At the bottom, there are "Save" and "Discard" buttons.

In this screenshot, we have created for Tina Robbins the job title of Production Manager. As you can see from this form, job titles are tied to departments. This means that to properly configure Odoo, you would need to create job titles across departments. Therefore, you do not necessarily want to name a job simply Manager. That would make it difficult, when looking at the list of job titles, to know with which department that manager may be associated.

You will notice that there is a place to enter the new employees that are to be expected and it defaults to 1. You can also see that the status of this job title shows **Recruitment in Progress** as well as a **Not Recruiting** button, which we will cover later in this chapter.



If you try to save the record you may get an error: **Error! You cannot create recursive hierarchy of Employee(s)**. It appears because Odoo knows Tina is in the production department and that the manager of that department should be Tina so it fills in the **Manager** field. Clear the **Manager** field to proceed.

Manager and Coach

The **Manager** and **Coach** fields in the employee screen can be used to specify any other employees that are already in Odoo. The manager is often called the supervisor in some companies and may be involved in approving the employees' timesheets, leave requests, performance appraisals, and so on. The coach is just an optional field that you could use to specify another relationship the employee has that is valuable to the position.

Employee Private Information

The **Private Information** tab on the employee screen contains the individual's private details pertinent to the HR department, such as home address, date of birth, and citizenship status:

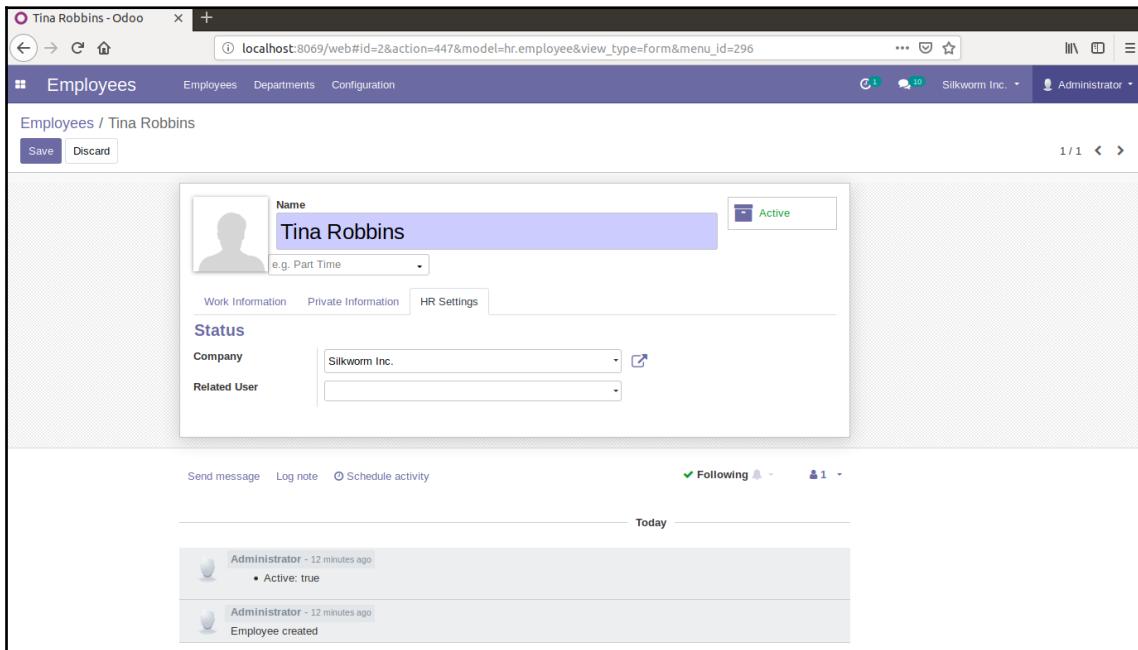
The screenshot shows the Odoo interface for managing employee data. The top navigation bar includes links for Employees, Departments, Configuration, and a notification badge for 10 messages. The main title is "Employees / Tina Robbins". Below the title are "Save" and "Discard" buttons. The main content area displays the "Private Information" tab for employee "Tina Robbins". The form is divided into several sections: "Citizenship & Other Information", "Contact Information", "Status", "Birth", "Work Permit", and "Education". The "Citizenship & Other Information" section includes fields for Nationality (Country, dropdown: United States), Identification No (4432), Passport No (empty), Bank Account Number (dropdown), Status (Gender: Female, Marital Status: Single, Number of Children: 0), Work Permit (Visa No, Work Permit No, Visa Expire Date), and Employee Documents. The "Contact Information" section includes fields for Private Address (dropdown), Emergency Contact (dropdown), Emergency Phone (dropdown), and Km home-work (0). The "Birth" section includes fields for Date of Birth (06/12/1967), Place of Birth (dropdown), and Country of Birth (dropdown). The "Education" section includes fields for Certificate Level (Master), Field of Study (dropdown), and School (dropdown).

The **Nationality (Country)** field allows you to select from the entire country listing that comes pre-loaded in Odoo. Typically, the **Identification No** field would be used for an employee badge. Odoo includes a **Passport No** on the form, which may be required in some cases where a company is required to report citizenship information to the government.

If you do decide to enter a home address for the employee, you will be taken to another screen. Near the bottom of the form, you have the ability to specify the **Gender**, **Marital Status**, and **Date of Birth** for the employee.

HR Settings

On the **HR Settings** page, the **Related User** field will allow you to associate the employee with an existing user account in Odoo. Simply select the user from the pop-up list and choose which user you want associated with the employee. It is also possible to add users on the fly by choosing **Create** from the **Related User** popup:

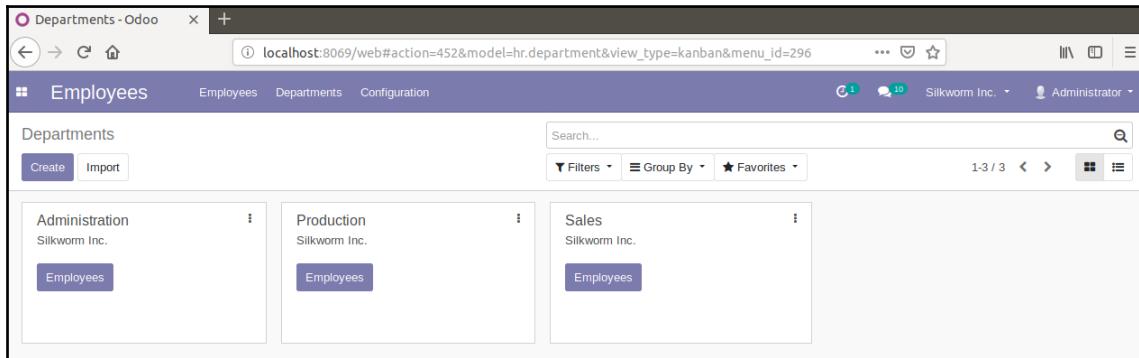


If the employee is not a user in the system, you can simply leave this field blank as we have in the preceding screenshot.

Managing departments

Earlier, we saw how we can add a department on the fly as we are entering an employees' details. Odoo also provides a dedicated **Departments** dashboard that lets you see and add departments to the system.

Click on the **Departments** link to bring up the form:

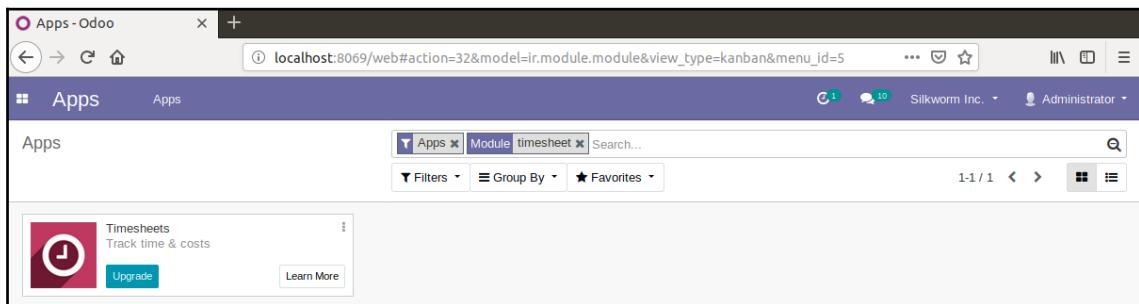


The screenshot shows the 'Departments - Odoo' page. At the top, there are buttons for 'Create' and 'Import'. Below that, there are three cards representing departments: 'Administration' (Silkworm Inc.), 'Production' (Silkworm Inc.), and 'Sales' (Silkworm Inc.). Each card has a blue 'Employees' button at the bottom. The interface includes standard Odoo navigation elements like back/forward, search, filters, and a header with user information.

The default installation of the **Employee Directory** application comes with two departments, **Administration** and **Sales**. We added the **Production** department in the previous section when we added Tina Robbins as an employee. Clicking on the **Employees** button on either department will take you to the appropriate list of employees for that department.

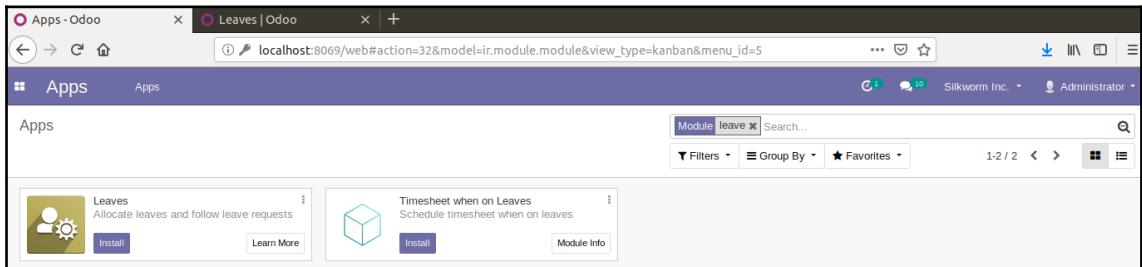
Timesheets

Odoo allows you to install an HR app that will allow you to track employee time and attendance. **Timesheets** are most useful when you have jobs that require you to account for employees' work hours and assign those hours to projects or customers. Unfortunately, in Odoo 12, the **Timesheets** application has been removed from the Community edition and is only available if you purchase Odoo Enterprise. This is indicated by the **Upgrade** button:



The screenshot shows the 'Apps - Odoo' page. In the search bar, 'Module timesheet' is typed. Below the search bar, there is a card for the 'Timesheets' module, which is described as 'Track time & costs'. The card features a clock icon and a 'Upgrade' button. The interface includes standard Odoo navigation elements like back/forward, search, filters, and a header with user information.

However, we can still utilize some of the timesheet features; they are just tucked away a little bit within the **Timesheet when on Leaves** module. Clear the **Apps** filter and search for **Leaves** to locate this module:



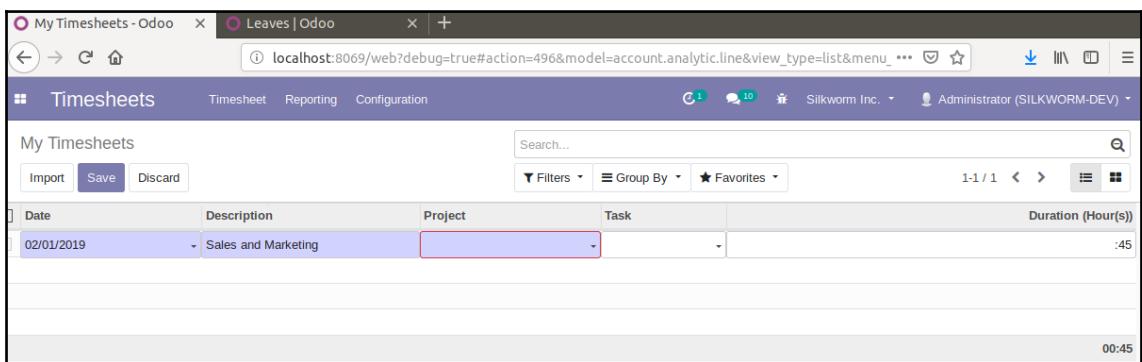
Once you have installed the **Timesheet when on Leaves** module, the menu bar will be expanded to include two new main menus, called **Timesheets** and **Leaves**. Currently, after installing the **Timesheets** application, Odoo just takes you to the standard **Discuss** menu just as if you have logged in.



When you install **Timesheet when on Leaves**, several dependencies will also be installed. One of them is the **Leaves** application. A little later in this chapter, we will look at the features in that application.

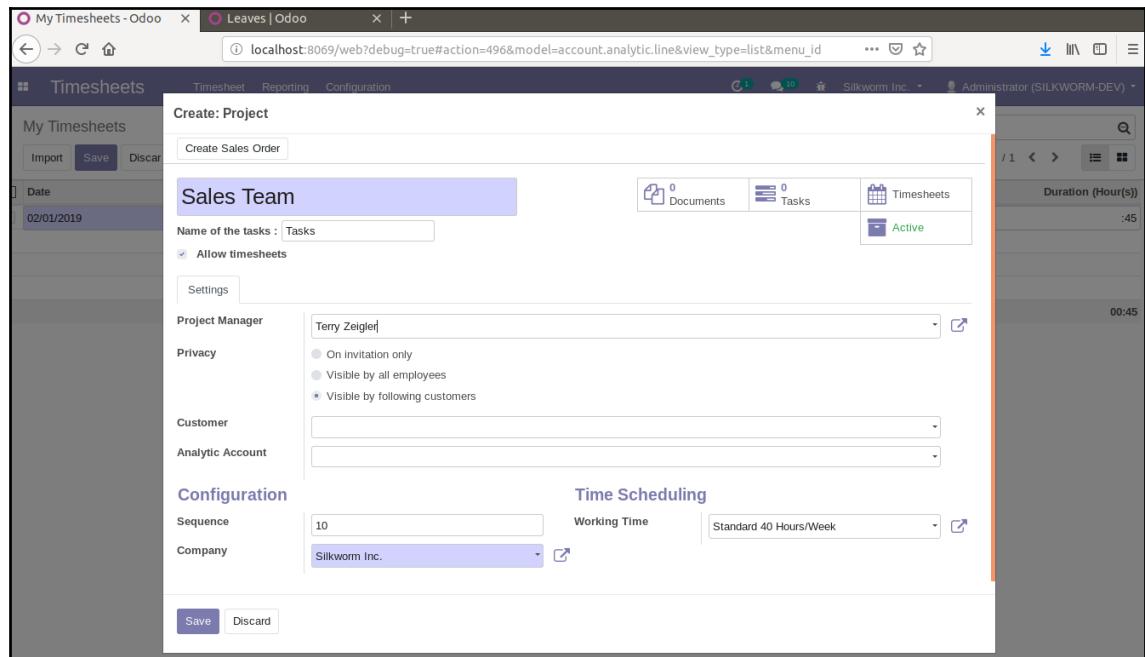
Click the **Timesheets** menu to pull up the application.

To begin recording employee activities, click on **My Timesheets**, then click on the **Create** button:



As you can see, we are presented with a form that allows you to enter the activities line-by-line. First, let's set the duration to 45 minutes. Each line has three required fields. The **Date** is filled in with the current date by default. You must then enter a **Description** and provide a **Project** to attach the activity to.

A **Project** is how Odoo ties the duration of the activity to the appropriate accounting entries. We will learn more about projects in the next chapter, *Chapter 9, Understanding Project Management*. For now, we will simply create a project to attach the activity to:



One thing you will notice is that you can assign a customer to the project as well. This can then allow you to more directly invoice customers based on a given project or activity. We are going to leave this blank for now and consider this an internal project for which we need to track the time. Save the new project and save the timesheet.

Now create another line and name the task Sales Department Report and set its **Duration** to 1 : 30. Change to kanban view and your timesheet should appear similar to following screenshot:

The screenshot shows the Odoo web interface for the Timesheets module. The top navigation bar includes tabs for Timesheets, Timesheet, Reporting, and Configuration. The main area is titled 'My Timesheets' and contains two tasks in a kanban view. The first task is 'Sales Team' under 'Sales and Marketing' with a duration of 00:45. The second task is 'Sales Team' under 'Sales Department Report' with a duration of 01:30. The interface includes standard Odoo navigation and search tools.

From this kanban view, you can view and edit a task by clicking on it once.

Leave management

In addition to managing and approving daily timesheets, it is also possible to install an Odoo HR application that will manage holidays, leave, and other information related to employee time off. When we installed the timesheet module, Odoo automatically installed **Leaves** as a dependency. You can confirm it is installed just by going to **Apps** and looking at it:

The screenshot shows the Odoo Apps module. The top navigation bar includes tabs for Apps, App Store, Updates, Update Apps List, and Apply Scheduled Upgrades. The main area is titled 'Apps' and lists the 'Leaves hr_holidays' application. It has a status of 'Installed' and a brief description: 'Allocate leaves and follow leave requests'. The interface includes standard Odoo navigation and search tools.

The primary purpose of the **Leaves hr_holidays** application is to provide an easy mechanism for employees to request leave and for their managers to approve or deny the request.

Creating a leave request

When you click on the **Leaves** menu option, you are taken to a monthly calendar that will show you your current leave requests. Naturally, if there were no prior leave requests made, or there are none for the current month, then the calendar is empty.

Click on a day in the calendar to tell Odoo to schedule a leave request beginning on that day:

The screenshot shows the Odoo Leaves module interface. At the top, there are tabs for 'Leaves' (selected), 'Overview', 'My Leaves', 'Managers', 'Reporting', and 'Configuration'. The URL in the browser is `localhost:8069/web?debug=true#id=&model=hr.leave&view_type=form&menu_id=304`. On the right, it shows 'Administrator (SILKWORM-DEV)' and other user information. The main area is titled 'All Leaves (February 2019) / New'. It has buttons for 'Save' and 'Discard'. Below that is a 'Reset to Draft' button. There are two tabs at the top right: 'To Approve' (selected) and 'Approved'. The form fields include:

- Leave Type:** Sick Leaves
- Duration:** 1.00 Days
- From:** 02/06/2019 To 02/06/2019
- Description:** Half Day
- Mode:** Employee
- Employee:** Tina Robbins
- Department:** Production
- Reported in last payslips:**

Below the form is a section for 'Comment by Manager' with a placeholder 'e.g. Report to the next month...'. At the bottom, there are buttons for 'Send message', 'Log note', and 'Schedule activity'. A 'Follow' button with 0 notifications is also present. A message at the bottom says 'Administrator - a minute ago Creating a new record...'.

As you can see, Odoo **Leaves** offers flexible options for setting up a leave schedule for employees.

Leave type

For our example, we have chosen **Sick Leaves** as the **Leave Type**. This implies that the employee is taking this leave with pay. Alternative leave types can be managed for reporting purposes.

Duration

When you change the **Duration** using the date range fields, Odoo will automatically re-calculate the **Days** field.

Mode

The **Mode** field deserves special explanation and dramatically changes the way in which this leave request is submitted. For our example, we are submitting the leave request for a single employee. By using the **By Employee** mode, you can submit leave requests that match all employees that share that same employee. This can be useful if you need to schedule leave for entire sets of employees.

Employee

This field lets you set the employee for whom leave is requested.

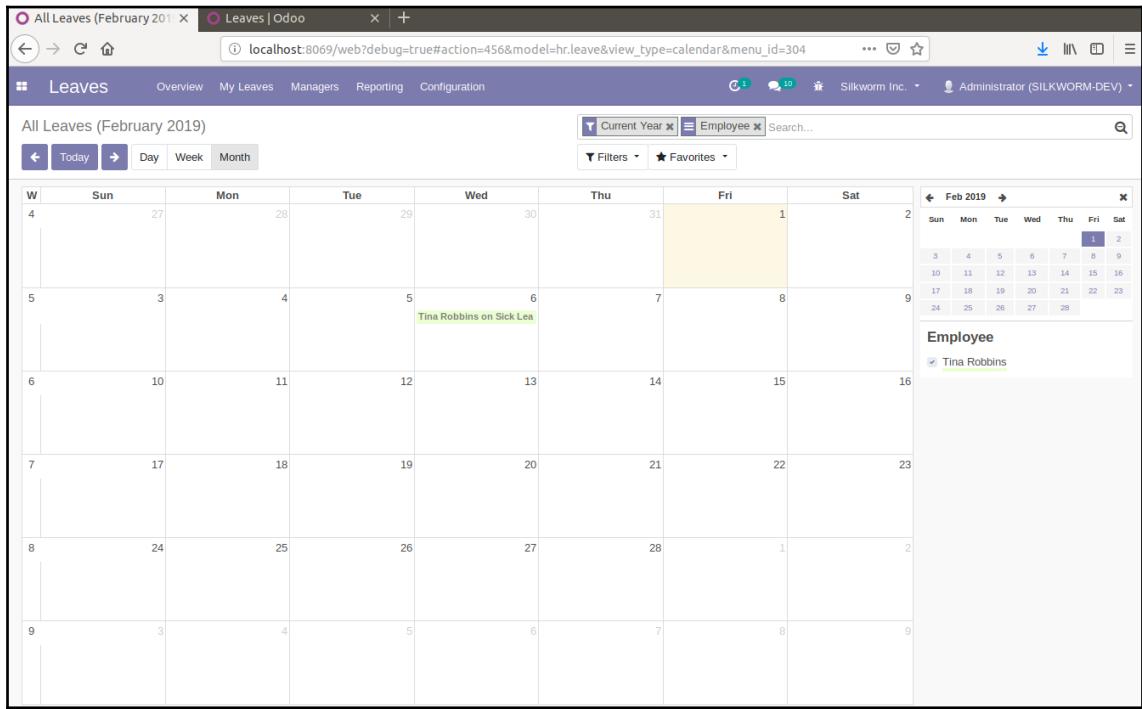
Department

This field lets you set the department for which leave is requested.

Submitting for approval

When requesting leave, clicking the **Save** button is all that is required to save the information and send it on to the assigned manager of the employee for approval.

Once you have saved the leave request, the form will close and you will see the updated calendar:



As you can see, our leave request is presented on the calendar with the appropriate information.

Approving leave requests

Clicking on the Leaves option in the Leaves to Approve section of the Human Resources menu pulls up the list of leave requests to approve. You may have to clear the My Department Leaves filter if the leave is not showing in the list. In our example, we see the leave request we have submitted for Tina Robbins:

Employee	Mode	Leave Type	Description	Start Date	End Date	Number of Days	Status	Reported in last payslips
Tina Robbins	By Employee	Sick Leaves	Sick Leaves	02/06/2019 00:00:00	02/06/2019 09:00:00	1.00	To Approve	<input checked="" type="checkbox"/>

To approve or deny a leave, simply click on a request and then choose **Approve** to approve the request or **Refuse** to deny the request. Now, in Odoo 12, you can approve a request by just clicking on the green check-mark on the far-right of the leave.

Recruitment process

Many HR departments can spend a great deal of time managing the recruitment process. Odoo provides an application that can help to organize the information and make it easier to keep track of the communication required when hiring new employees.

Install the **Recruitment** application as you have the other Odoo applications:



Note: In the preceding screenshot, you will notice **hr_recruitment** has been added to the end of the application title. This is because we are in developer mode, which we learned how to turn on in Chapter 6, *Configuring Accounting – Finance Options*. **hr_recruitment** is the technical name for the **Recruitment** application.

After the **Recruitment** application has been installed, you can access the application by selecting the newly added **Recruiting** menu.

You are then presented with the **Job Positions** form:

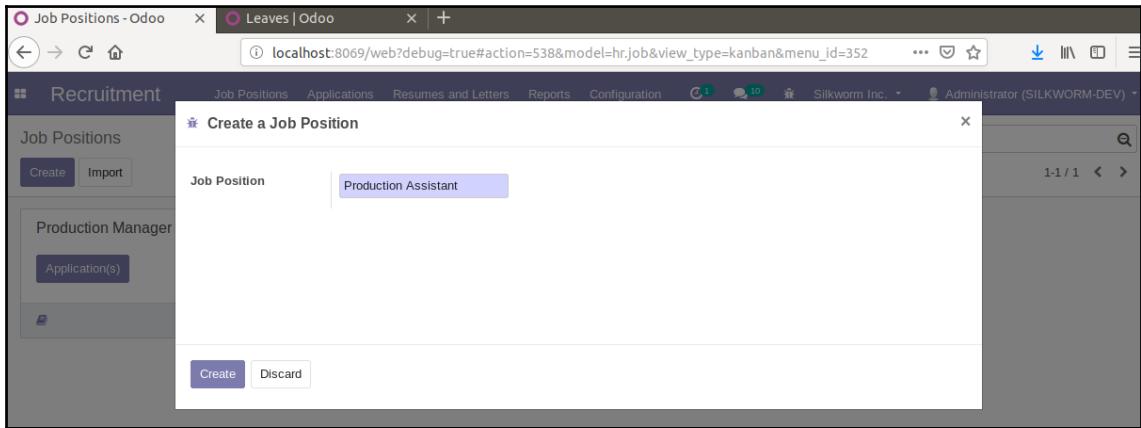
A screenshot of the Odoo web interface showing the 'Job Positions' kanban view. The title bar shows 'Job Positions - Odoo' and 'Leaves | Odoo'. The URL is 'localhost:8069/web?debug=true#action=538&model=hr.job&view_type=kanban&menu_id=352'. The top navigation bar includes 'Recruitment', 'Job Positions', 'Applications', 'Resumes and Letters', 'Reports', 'Configuration', and 'Administrator (SILKWORM-DEV)'. Below the navigation is a search bar and filter options ('Filters', 'Group By', 'Favorites'). The main area displays a single job position card for 'Production Manager'. The card has a blue header and contains the job title, a 'Create' button, and a 'Import' button. Below the title is a section labeled 'Application(s)' with a small icon. At the bottom of the card is a footer with a small icon and the text '0 Hired Employees'.

You will then see the **Production Manager** job position we added earlier in this chapter. Like the other kanban setups in Odoo, each position provides various options to color the panel and perform other common operations.

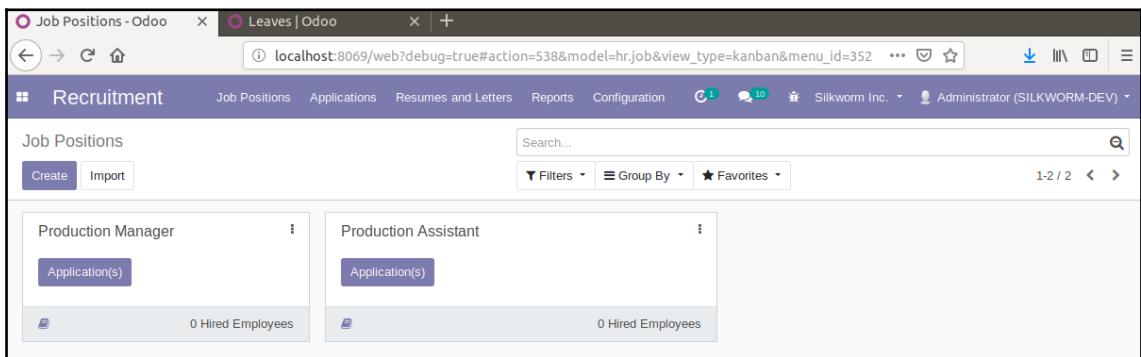
Recruiting for a new job position

Tina Robbins has been very busy in her position of **Production Manager**. It has been decided that there is a need to hire a **Production Assistant** to assist her in her duties. With the new recruitment application installed, we can now create a new job position and start the recruiting process.

Click on **Job Positions** under the **Recruitment** menu and click on the **Create** button:



Here, we have filled in the **Job Position** for our **Production Assistant**:

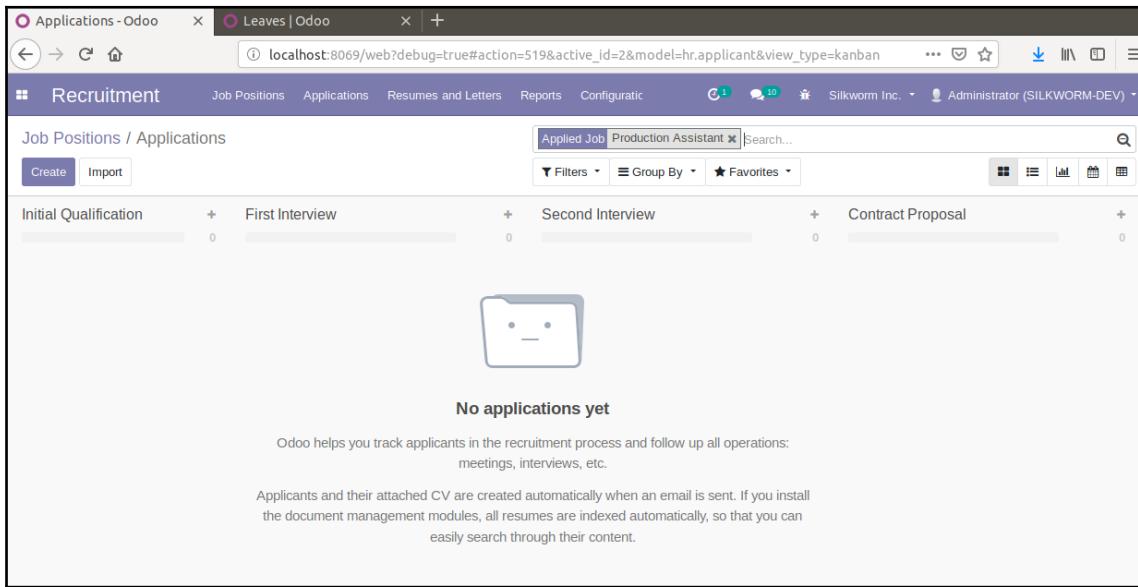


When we go back and look at the **Job Positions** in Odoo, we will find that the kanban view now displays details about the job position we created.

Creating an employment application

When a potential employee sends in an application, resume, or another trigger that will allow you to document their interest in working for your company, you create a recruitment application.

Click the **Application(s)** button for the **Production Assistant** to bring up the application list for the job position:



Because there are no applications, naturally, this list will start out empty. Also, notice that there is a filter applied that is restricting the applications to only those that are associated with the **Production Assistant** position.

Click on **Create** to add a new application:

The screenshot shows the Odoo HR Application form. At the top, there are tabs for 'New - Odoo' and 'Leaves | Odoo'. The URL bar shows 'localhost:8069/web?debug=true#id=519&active_id=2&model=hr.applicant&view_type='.

The main header includes 'Recruitment' and navigation links for 'Job Positions', 'Applications', 'Resumes and Letters', 'Reports', 'Configuration', and user information ('Administrator (SILKWORM-DEV)').

The form itself has a title 'Job Positions / Applications / New'. It contains several sections:

- Subject / Application Name:** Production Assistant - B. Nelson
- Applicant's Name:** Bob Nelson
- Contact:** bobnelson@examplemail.com
- Responsible:** Appreciation
- Email:** bobnelson@examplemail.com
- Phone:** 333-444-5555
- Medium:** Medium
- Mobile:** 222-333-4444
- Source:** Search engine
- Degree:** Bachelor Degree
- Referred By:** (empty)
- Job:** Applied Job: Production Assistant; Department: Production; Company: Silkworm Inc.
- Contract:** Expected Salary: 42000; Proposed Salary: 35000; Availability: 02/28/2019
- Notes:** Medical and Vacation; Medical Only

At the bottom right, there are buttons for 'Meetings' (0) and 'Documents' (0).

This form has a lot of fields to be potentially filled out. By default, the only required field for the application is the subject. The rest of the information can be collected throughout the recruiting and interviewing processes.

Hiring employees

Let's go ahead and hire this Bob Nelson guy. Thankfully, Odoo's **Recruitment** module will create the employee for us by simply clicking the **Create Employee** button near the top of the form:

The screenshot shows the Odoo web interface for the Recruitment module. The title bar indicates the current view is 'Job Positions / Applications / Production Assistant - B. Nelson'. The main content area displays the application details for 'Production Assistant - B. Nelson'.

Contact Information:

- Email: bobnelson@examplemail.com
- Phone: 333-444-5555
- Mobile: 222-333-4444
- Degree: Bachelor Degree

Job Information:

- Applied Job: Production Assistant
- Department: Production
- Company: Silkworm Inc.

Responsible Appraisal:

- Medium Source
- Referred By: Search engine

Contract Details:

- Expected Salary: 42,000.00 + Medical and Vacation
- Proposed Salary: 35,000.00 + Medical Only
- Availability: 02/28/2019

Application Summary:

Meetings: 0 Documents: 0

Looking at the **Employees** list, we can now see that Bob Nelson is an employee in the **Production** department with the title of **Production Assistant**:

The screenshot shows the Odoo web interface for the Employees list. The title bar indicates the current view is 'Employees'.

Employees List:

- Administrator
- Bob Nelson (Production Assistant)
- Tina Robbins

Using Odoo for recruiting and hiring employees is both easy and straightforward. Putting in the time to set up this process can improve your hiring process and allow you to keep all important information in one place.

Summary

In this chapter, we examined the various HR applications available in Odoo. We installed the base **Employee Directory**, followed by applications that managed time and attendance, as well as leave requests. Finally, we installed an application that allowed us to manage the recruiting processes of new employees. We walked through completing an employment application and finally, turning the potential applicant into an employee.

In the next chapter, we will look at the **Project Management** application in Odoo and how it can be used to improve service quality for customers. **Project Management** allows you to organize the simplest projects though to complex projects involving multiple tasks. Furthermore, you can even track the time spent on projects and display project information in a variety of graphical formats to make it easier to track your deadlines.

9

Understanding Project Management

In this chapter, we will explore a very flexible application that allows you to manage projects and tie them into other applications in Odoo. The Project Management application allows your company to manage project stages, assign teams, and even track time and job costs related to projects. Analytical accounting features give you even greater control over how project costs can be tied to your company's general ledger.

This chapter covers the following topics:

- Discovering the various uses for Project Management
- Linking projects with customer accounts
- Assigning teams to projects
- Creating custom project stages
- Adding, assigning, and organizing tasks
- Tying into analytical accounting and employee timesheets

What's new in Odoo 12?

Project Management in Odoo 12 has undergone some minor interface changes that simplify a number of operations. In addition to this, Project Management now ties into Odoo 12's new activity menu.

The basics of Project Management

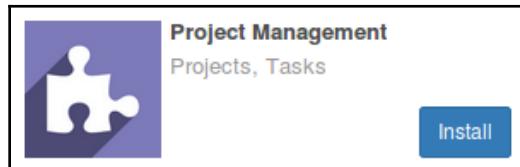
Depending on your industry and the types of projects you may encounter, the Project Management application can be set up to manage independent projects. Alternatively, it can be configured to manage projects related to customers or sales orders. With additional modules, it is possible to tie the Project Management application into virtually any aspect of Odoo.

For example, you could simply use the Project Management application to track the various stages and tasks involved with a company event. Who is going to be responsible for finding the location? When will you need to order the invitations? Who is going to set the agenda? When is an employee going to go and pick up the sound system? In this instance, the Project Management application is simply being used to track a single project that is not associated with the customer.

In other instances, you may wish to utilize the Project Management application to track projects that are organized around your customer records. A common example would be a construction firm. After assigning the project to a customer, you can track various stages of the project life cycle. Employees can be assigned tasks and, using the Odoo messaging system, you can share project details with your customers. It is in this configuration that the Odoo Project Management application can add real value to an Odoo installation, and, by doing so, provide better integration with your accounting system with less effort than a standalone project management tool.

Installing the Project Management application

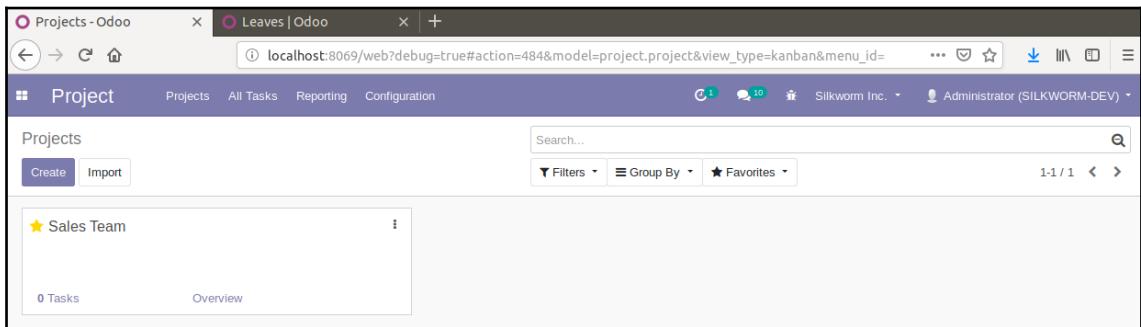
To access the project managing features, if you have not followed along in Chapter 8, *Implementing the Human Resources Application*, you will need to install the **Project Management** application. In Odoo 12, **Project Management** is automatically installed when you install the **Timesheet** application:



If **Project Management** is not installed, go to the **Settings** menu and install the **Project Management** application using the same process as the previous Odoo applications.

Understanding the Project dashboard

After installing the **Project Management** application, the screen will be refreshed, and you will see the **Project** dashboard. The dashboard provides an overview of the active projects and comes with the project we added in the previous chapter, **Sales Team**:



We can see in the summary that the project we created in the previous chapter has **0 Tasks**, as we never added any tasks to the project.

While this project was useful as an internal example of using a project to time various employee activities, let's explore how we can use the Project Management application to manage various tasks for a customer.

A real-world project example for a customer

As in other chapters, we will use a real-world example to demonstrate the functionality of Odoo's Project Management application. In the silkscreen industry, it is common to have extremely large projects that span many different types of apparel and print designs. For this example, we are going to create a project to manage the creation of an entire line of sports jerseys for an organization called **Lil League**.

In defining our project, it is important to look at the scope of our project and why it will be valuable to use the project manager to organize the various tasks involved. With our Lil League organization, we have multiple teams that can vary in logo design for the team, the number of players, the sizes of the apparel required, and the printing of different players' numbers and even players' names. There are often multiple deadlines to manage and a number of people that may need to approve various phases of the project as they are completed. Using the Project Management application, we can better track this information and tie it into sales orders and other Odoo functions.

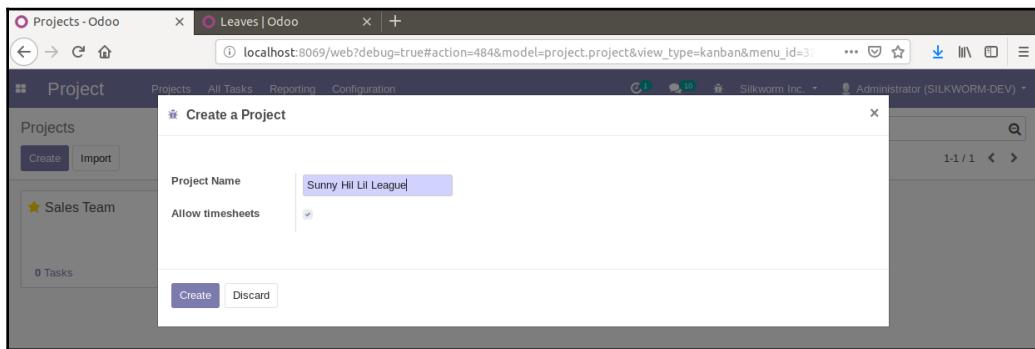
Creating our first project

After the **Project Management** application has been successfully installed, we can go to the **Project** application and create a new project.

We will follow these basic steps:

1. Create a new project record.
2. Give a name to the project.
3. Assign the project to a specific customer.
4. Assign team members to the project.

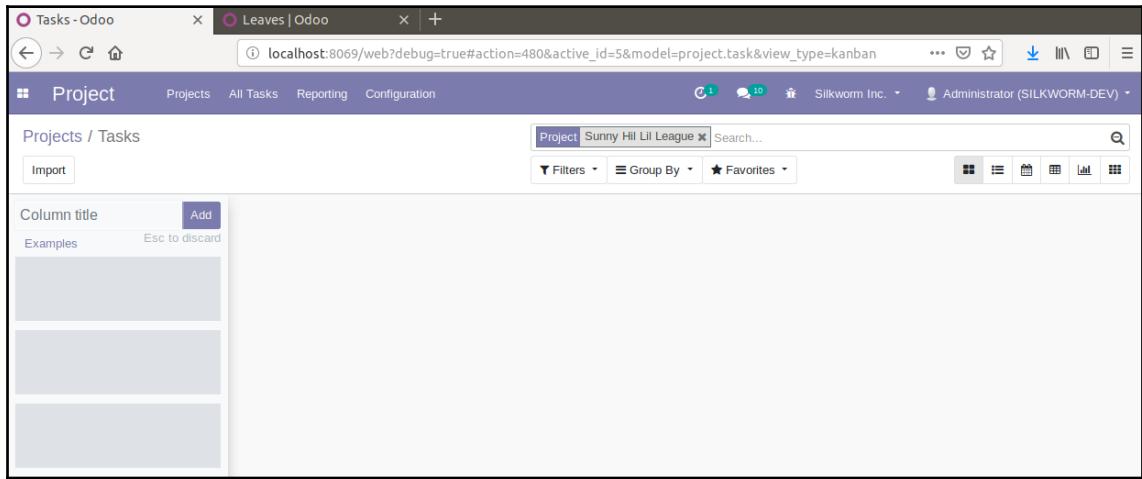
To begin, under the **Project** application, select **Dashboard** in the left-hand menu, and then click on the **Create** button:



In previous versions of Odoo, you were taken to the **Project** editor after clicking on **Create**. Like many processes in Odoo 12, this is now greatly simplified, and you are presented with a very simple wizard that prompts you for a **Project Name** and the option to **Allow Timesheets**. For our example, we will use Sunny Hill Lil League as the name for our project.

As you may expect, checking the **Timesheets** checkbox will allow you to associate timesheets with the project in the same way that we did in the previous chapter. We will go ahead and leave this checked so that our project can manage the labor that is spent on the project, and then bill it to the customer when required.

Once you create the project, Odoo will refresh and immediately take you to the **Project** editor:



When you first create your project, there is not much to it. However, we can quickly create our project structure by adding project stages to organize the tasks in our project.

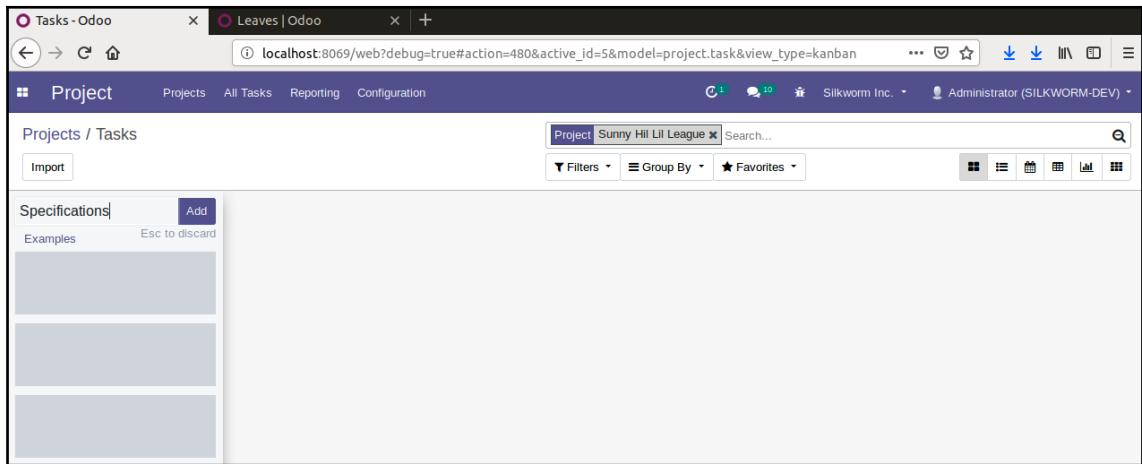
Understanding project stages

Project stages allow you to track a given task through different phases of its completion. These stages will often vary from project to project, depending on the types of tasks involved. For example, a project involving software development will likely have different task stages than one that involves planning a real-estate development seminar.

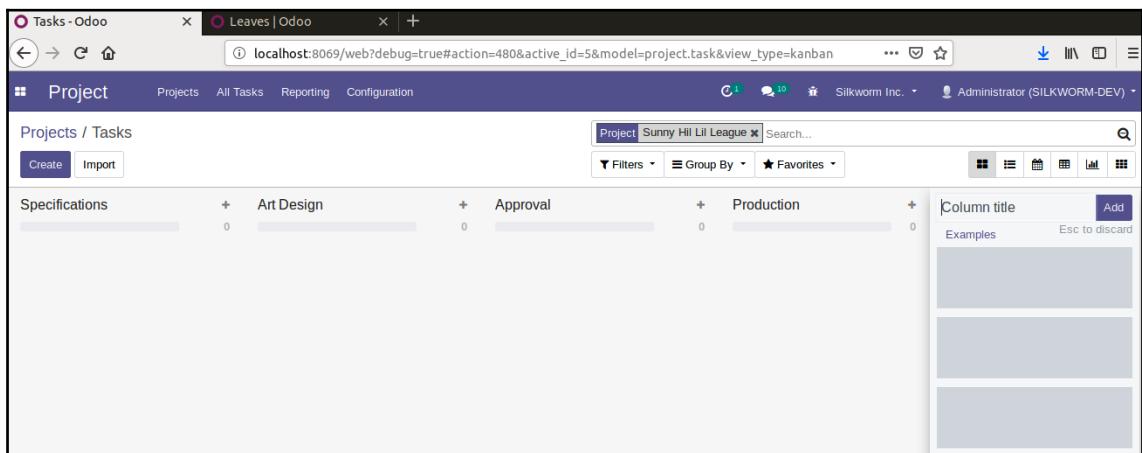
In Odoo 12, a new project does not provide default stages for your project. Fortunately, it is very easy to create new stages. Let's go ahead and create some stages for our new project.

For our example, we are going to create stages for **Specifications**, **Art Design**, **Approval**, and, finally, **Production**.

Simply type in the stage, in this case, **Specifications**, and then click on the **Add** button:



The screen will refresh to show the new column added. Use the same process to add the additional stages to the project:

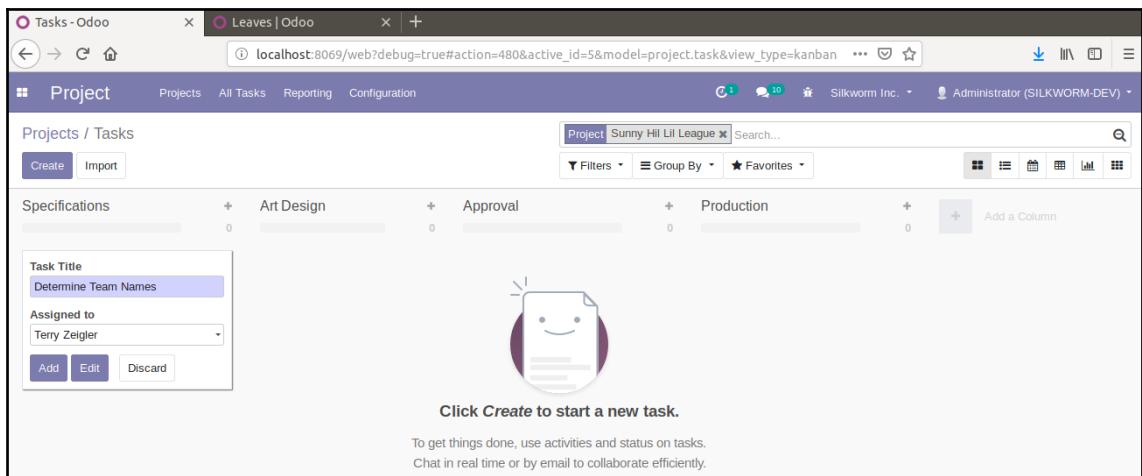


When you are finished adding, your project stages should look like the preceding screenshot. Now we are ready to begin adding tasks to our project.

Defining project tasks

The main unit for tracking the various activities involved with a project is a project task. Odoo provides a quick way for you to add a task to the project by clicking on the large plus button underneath the appropriate project stage.

Click on the plus button inside the **Specifications** stage, and then create a task for Determine Team Names:



In the same way as adding a stage, you simply enter the title of the task and then hit the **Add** button. The screen will then refresh, and you will automatically be prompted to add the next task. Once you have created the task, you can click on the task to open it, or you can use the menu on the task to pull the task up to edit it. Once you have opened the task, you can now provide additional details.

Here, we'll define additional aspects of the task such as:

- The name of the task (which is required)
- The **Project** to which the task is assigned
- The **Deadline** date of the task
- The responsible party **Assigned to** the task
- Any **Tags** you would like to associate with this task
- A **Description** of the task

For our example, we have filled out the task, as shown in the following screenshot:

The screenshot shows the Odoo project task form for the task 'Determine Team Names'. At the top, there are tabs for 'Specifications', 'Art Design', 'Approval', and 'Production', with 'Specifications' being the active tab. Below the tabs, there are fields for 'Project' (set to 'Sunny Hill Lil League'), 'Assigned to' (set to 'Terry Zeigler'), 'Deadline' (set to '02/12/2019'), and 'Tags' (set to 'Information Collection'). A rich text editor is present for the task description. A note at the bottom of the editor states: 'This initial specification will determine the team names required from the project. It is **VERY IMPORTANT** to double check the names and confirm with the client before proceeding.' At the bottom of the form, there are buttons for 'Send message', 'Log note', and 'Schedule activity', along with a 'Following' button and a notification icon.

At the top of the form, you will see all the project stages, with the current stage highlighted in blue. In this example, **Specifications** is the currently selected stage. When in edit mode, you can click on these stages to directly assign the task to a given stage. This can be changed as the project progresses, so you are not locked into keeping a task assigned to the same stage throughout the project.

One of the important aspects of good project management is assigning responsible parties for each task. The **Assigned to** field allows you to specify who is ultimately responsible for the completion of the task.

The **Tags** field can be valuable for better tracking and organizing tasks. In our example, we have defined an **information collection** tag. This tag can then be assigned to any task that is related to collecting data regarding the project.

Creating additional tasks

For our real-world example, we are going to define several tasks at various stages. When creating your own list of stages, think about the flow and order of your tasks. Ideally, a task will move from left to right and the stages are completed. For our example, our tasks include the following:

- Specifications about the details of the project
- Creating the Art Design
- Approval of the design
- Producing the product

Now that we have created generic tasks for the entire project, let's go ahead and assume that we have received a draft list of team names. This will allow us to track the progress of each team as it goes through the various stages in the project. For our example, we will use the following team names and leave them in the **Specifications** stage to start:

- Bulldogs
- Falcons
- Wolf Pack

After entering our **Tasks** and assigning them to the various stages, it should look something like this:

The screenshot shows the Odoo Kanban interface for a project titled "Sunny Hill Lil League". The interface is divided into four main stages: "Specifications", "Art Design", "Approval", and "Production". Each stage has a corresponding horizontal bar with a count of 1. Below each bar is a card representing a task. The "Specifications" stage contains one card for "Determine Team Names" (due 02/12/2019). The "Art Design" stage contains one card for "Falcons" (due 02/26/2019). The "Approval" stage contains one card for "Bulldogs" (due 02/21/2019). The "Production" stage contains one card for "Wolf Pack" (due 02/18/2019). The top navigation bar includes links for "Projects", "All Tasks", "Reporting", and "Configuration". The top right corner shows the user is an "Administrator" on the "SILKWORM-DEV" instance.

The Kanban interface allows you to pick up a task and move it to another stage, or reorder it within a current stage, by just clicking and dragging. You can reorder the stages by clicking on the title of the stage and dragging it to your desired position.

Additionally, note that we have added in deadline dates to the tasks. We can then view the tasks in the calendar view for the project.

Click on the calendar icon on the right-hand side of the form to bring up the calendar view:

The screenshot shows the Odoo Project module interface. At the top, there are tabs for 'Tasks (February 2019)', 'Leaves | Odoo', and a URL bar indicating the current page is 'localhost:8069/web?debug=true#action=480&active_id=5&model=project.task&view_type=calendar'. Below the header, there's a navigation bar with links for 'Project', 'Projects', 'All Tasks', 'Reporting', and 'Configuration'. On the far right, it shows 'Administrator (SILKWORM-DEV)' and other user information.

The main area is titled 'Projects / Tasks (February 2019)'. It features a large calendar grid for February 2019. The days of the week are labeled at the top: Sun, Mon, Tue, Wed, Thu, Fri, Sat. The dates are numbered from 1 to 28. Specific tasks are highlighted with green bars: 'Determine Team Names' (Feb 11-12), 'Wolf Pack' (Feb 17-18), 'BullDogs' (Feb 20-21), and 'Falcons' (Feb 25-26). To the right of the calendar, there's a sidebar with a smaller calendar for 'Feb 2019' and a section titled 'Assigned to' which lists 'Administrator' and 'Terry Zeigler'.

The calendar view allows you to see and organize your tasks. At the top, you have three buttons that allow you to change the view between **Day**, **Week**, and **Month**.

Completing project tasks

As you complete project tasks, you can bring them up and hover your mouse over the **Active** button. It will then change to **Archive** as you hover over it. When you click on the button, the task will be archived. This method will remove the task from the project entirely and hide it from view.

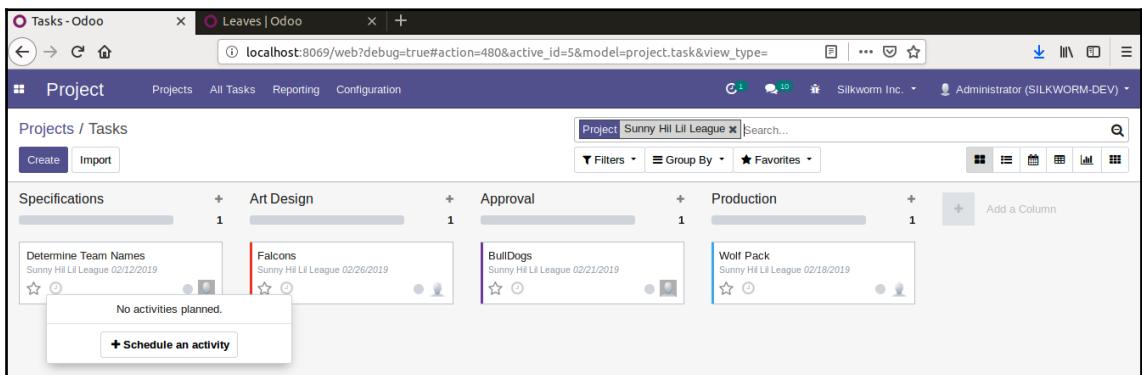


If you ever need to see tasks that have been archived, there is a search view already created for **Archived** that you can apply to your searches. Alternatively, you can use the advanced filter skills that you will learn about in chapter 10, *Creating Advanced Searches and Dashboards*, to set a filter that displays records in which active records are false.

Previously in Odoo, there was a **Done** status that was provided by default. If you need to keep track of completed tasks and not have them fall off the project, you will need to set up a **Done** or **Completed** stage for your project.

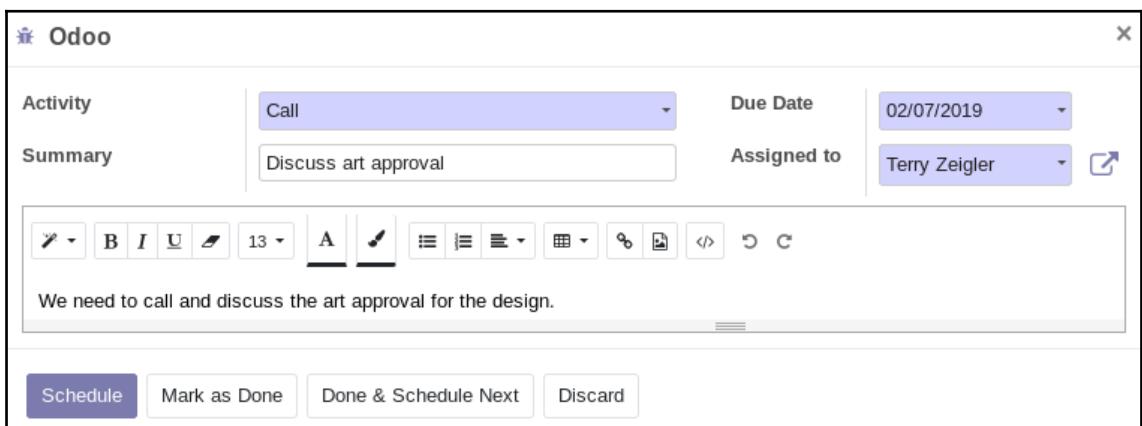
Scheduling an activity

In addition to managing task deadlines, you can also directly schedule activities related to a project task. When you schedule an activity, it can then be centrally managed through Odoo 12's **Activities** menu. Pull up the task you wish to schedule an activity for, and then click on the **Schedule an activity** link to pull up the form to schedule the activity. You can find this link just under the description of the task:



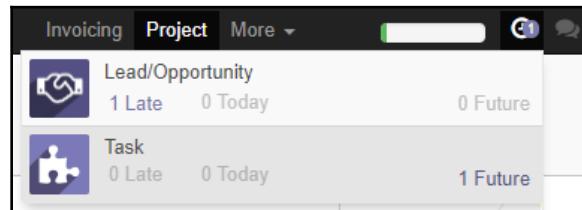
The screenshot shows the Odoo Project / Tasks interface. At the top, there are tabs for Projects, All Tasks, Reporting, and Configuration. The current view is on the Projects / Tasks tab. A search bar at the top right contains the text "Project Sunny Hill Li League". Below the search bar are buttons for Filters, Group By, and Favorites. The main area displays a grid of tasks under four categories: Specifications, Art Design, Approval, and Production. Each category has a count of 1. The "Art Design" category contains two items: "Falcons" and "Bulldogs". The "Production" category contains one item: "Wolf Pack". Each task item includes a star icon, a date, and a small profile picture. At the bottom left of the grid, there is a message: "No activities planned." Below this message is a button labeled "+ Schedule an activity".

Now, you can enter the details for the activity you wish to schedule. In this example, we have scheduled a call to discuss the approval of the art for a specific phase of the project:



The screenshot shows the Odoo Activity scheduling form. The top section has fields for Activity (set to "Call"), Due Date (set to "02/07/2019"), and Assigned to (set to "Terry Zeigler"). Below these are rich text editing tools. A text area contains the note: "We need to call and discuss the art approval for the design." At the bottom, there are four buttons: "Schedule", "Mark as Done", "Done & Schedule Next", and "Discard".

When you click on **Schedule**, you will then be able to see the task under the **Activities** menu:

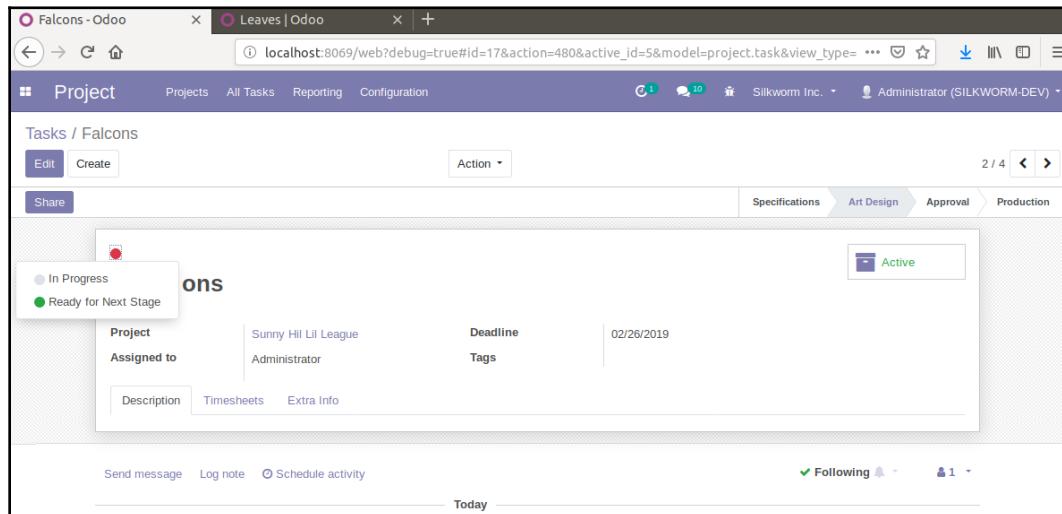


Here, we can see how the **Activities** menu consolidates your various tasks. In one glance, you can see the activities across all Odoo applications.

Monitoring task completion

Odoo uses a simple set of flags to determine whether a task is ready to move between stages. In the top-left corner, you will find a small round icon that allows you to specify whether the task is ready to go to the next stage (green), or whether the task is, for some reason, blocked from moving to the next stage (red).

So, for example, when a task moves to the approval stage, the person responsible for performing that task can choose to mark it red. This will tell anyone who is looking at the task that it is being held up:



Here, we have marked an approval task as blocked, so the icon shows red. As you can see, you can set it to gray to show that it is in progress (the default setting), or set it to green in order to specify that the task is **Ready for Next Stage**.

Using these flags, you can see at a glance on the Kanban view whether a task is ready to move forward, or whether it requires special attention.

Calculating project costs and time

To assist the process of calculating project time and costs, you can install the human resources application, **Timesheets**. This application lets you create timesheets for your employees that let you specify the number of hours worked per day. If you have been following along, this application should have already been installed in Chapter 8, *Implementing the Human Resources Application*.

If you click on the **Timesheets** page on a given task, you will get a list view where you can create new timesheet entries. In this example, we have attributed one hour to **Determine Players Names** in the **BullDogs** task:

The screenshot shows the Odoo Project module interface. The main navigation bar includes 'Project', 'Projects', 'All Tasks', 'Reporting', and 'Configuration'. The current view is 'BullDogs' under 'Projects / Tasks / BullDogs'. A modal window is open for the 'BullDogs' task. The task details include:

- Project:** Sunny Hill Lil League
- Assigned to:** Terry Zeigler
- Deadline:** 02/21/2019
- Description:** Determine the names & numbers
- Planned Hours:** 04:00
- Progress:** 25%
- Timesheets:** Tab selected, showing a single entry for Bob Nelson with a duration of 01:00.
- Extra Info:** Not visible in the screenshot.

At the bottom of the modal, status indicators show 'Hours Spent: 01:00' and 'Remaining Hours: 03:00'.

You can learn more about using timesheets and analytical accounting in Chapter 8, *Implementing the Human Resources Application*.

Summary

In this chapter, we examined the Project Management application by taking a look at the project dashboard. We then created an example of a real-world project involving our Lil League organization. After setting up our project and assigning team members, we defined the various stages involved in completing the project. With the stages defined, we were able to go through and assign various tasks to the stages, along with their dates of completion. Finally, we looked at the various ways you can view the tasks and how you can complete them.

In the next chapter, we will explore how you can create advanced searches and custom dashboards in Odoo. As a company uses its system from day to day, the amount of collected data can grow quite rapidly. Being able to locate pertinent records in a speedy manner is vital for optimum business operation. We'll discover how to utilize all of the handy searching, filtering, and dashboard presentation tools that are at our disposal within Odoo.

10

Creating Advanced Searches and Dashboards

In this chapter, we will cover advanced searching, custom filters, and dashboards. We'll begin by looking at how Odoo searches the various datasets within the system. Then, we'll explore more advanced searching options and discuss how you can save these filters so that they can be easily accessed when you need them. Finally, we will discuss the Odoo dashboard's capabilities and how we can improve its usability for users.

The following topics will be covered in this chapter:

- Identifying a user's search requirements
- Understanding default filters versus custom filters
- Grouping items in a list
- Setting and saving advanced search conditions
- Creating dashboard content and layouts

Determining the search requirements for your business

One of the tasks that can often be frustrating and time-consuming for users is trying to find the information they need. When datasets are small and simple, there is not much of an issue. As the number of records in the system grows, it can become increasingly hard to find information.

When implementing an **enterprise resource planning (ERP)** system, you will want to take the time to work with users and get familiar with the data they use each day. If you are working with a purchasing system that only produces an average of 10 purchase orders a day, you will have few concerns raised over advanced searching in that application. However, if you have 20 purchasing agents cutting 450 purchase orders a day, it will be critical that the users have a firm grasp on the search functionality of the system. Trying to locate a specific order can be like trying to find a needle in a haystack.



Take the time to sit with users and watch them use the system. Often, users will need to look up the same types of data repeatedly in their daily interactions with the system. These are the activities that you will want to set up custom filters for, and perhaps even include on the user's dashboard.

Fortunately, Odoo offers a robust searching mechanism, as well as the ability to create dashboards for displaying information that the user may need to look at frequently.

In this chapter, we will create a new database with demonstration data so that you can better see the searches in action.

Creating a database with demonstration data

It can often be valuable to test certain features in Odoo without having to enter a lot of data. Odoo offers an option to populate a database with demonstration data when you create it. Since this chapter is focused specifically on searching for and displaying data, we will load up a database with the sample data provided by Odoo.

Accessing the database manager

While it is possible to access the database manager by clicking on a link on the login screen, there are times when that link is not available. One reason for this could be that the website application has been installed and the **Manage Databases** link is hidden from the home page.



Sometimes, if your Odoo server is throwing internal server errors, or you are having other problems with your database, you can resolve the issue and make backups of your data by going directly to the database manager.

To access the database manager directly in the default installation of Odoo, you can use <http://localhost:8069/web/database/manager>.

You will then be presented with the database manager screen:

The screenshot shows a Firefox browser window with three tabs open: 'Odoo', 'Odoo', and 'Yahoo'. The 'Odoo' tab is active and displays the Odoo logo. Below the logo is a yellow warning box containing the text: 'Warning, your Odoo database manager is not protected. Please [set a master password](#) to secure it.' Underneath the warning box is a table with one row. The first column contains the database name 'SILKWORM-DEV'. To the right of the database name are three buttons: 'Backup' (blue), 'Duplicate' (grey), and 'Delete' (red). Below the table are three blue buttons: 'Create Database', 'Restore Database', and 'Set Master Password'. At the bottom of the browser window, there is a Mozilla Firefox status bar message: 'Mozilla Firefox seems slow... to... start.' followed by two links: 'Learn How to Speed It Up' and 'Don't Tell Me Again'.

Naturally, you will need to change the server address and port to match your Odoo installation, as shown in the following steps:

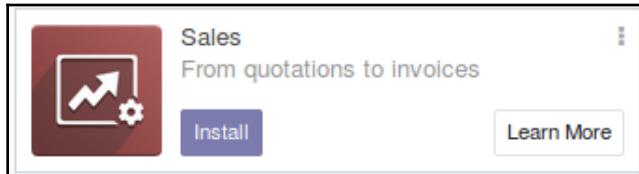
1. Click on the **Create Database** button to create a new database.
2. When creating our database, we check **Demo data** so that our database is prepopulated, making it easier to present example search techniques:

The screenshot shows a modal dialog titled "Create Database". It contains fields for entering database details and preferences. The fields include:

- Database Name: AdvancedSearching
- Email: admin@youremail.com
- Password: (obscured)
- Phone number: (empty)
- Language: English
- Country: United States
- Demo data: A checked checkbox.

Below the fields, there is a note: "To enhance your experience, some data may be sent to Odoo online services. See our [Privacy Policy](#)." At the bottom right of the dialog is a blue "Continue" button.

3. After you click on **Continue**, Odoo will set up the new database. So now, we need an application to work with, and we will install the **Sales** application:



When you run the **Sales** application this time, you will see the demo data that consists of several quotations and sales orders in various stages of completion.

New demonstration data in Odoo 12

Odoo 12 now comes with new and improved demonstration data. This data is for a fictional company that sells both office furniture and interior design services, thereby providing examples of each type of product—that is, consumable and service products. Even the **Discuss** area under each quotation, sales order, invoice, and product already contains a log of when each item was created and by whom. In short, the demo data for Odoo 12 is more well thought out, thorough, and logical than in prior versions of Odoo.

In Odoo 12, you can now load demo data even after your database has been created. Simply go to **Settings** and activate the developer mode. Then, return to **Settings** once more, and you'll see a link in the bottom-right corner that reads **Load demo data**. It will warn you that, once added, the demo data cannot be removed. So, make sure you never add demo data to an existing database that you wish to use for serious purposes. (On a humorous note, in the demo data under settings for **Your Company**, you'll find a familiar address if you were a fan of the American TV mockumentary *The Office*. Yes, that is the street address for *Dunder Mifflin Paper Company*.)



Searching in Odoo

Odoo provides a standardized search box in the top-right corner of most forms. Depending on the menu item, some forms come with predefined filters already set for the list.

If you navigate to the **Sales** application, go to the **Order** menu, and select **Customers**, you can see the search box in the top-right corner with the **Customers** filter preassigned:

The screenshot shows the Odoo web interface for the Sales application. The top navigation bar includes links for Sales, Orders, To Invoice, Products, Reporting, and Configuration, along with a user dropdown for Mitchell Admin. The main content area is titled 'Customers' and displays a kanban view of six customer records. Each record card includes the company name, address, email, and a small profile picture. A search bar at the top right is set to 'Customers'. Below the search bar are buttons for 'Filters', 'Group By', and 'Favorites', and a page navigation section showing '1-6 / 6'. The records listed are:

Customer	Type	Address	Email	Photo	Count
Azure Interior	Services	Jonesboro, United States	azure.interior24@example.com		\$ 1
Deco Addict	Vendor / Desk Manufacturers	Franklin, United States	deco.addict82@example.com		\$ 2
Gemini Furniture	Consulting Services	Tampa, United States	gemini.furniture39@example.com		\$ 13
Lumber Inc		Asheville, United States	lumber-inv92@example.com		\$ 1
The Jackson Group		Auburn, United States	jackson.group82@example.com		\$ 2
YourCompany, Joel Willis		Bayonne, United States	joel.willis63@example.com		\$ 2

Some lists open with a predefined filter that will limit the primary dataset. In the previous screenshot, you can see that the list view, **Customers**, has a customer filter applied by default when you open the form. Odoo stores customers, vendors, and employees in the same central database table. The customer filter prevents those other types of data, such as vendors or employees, from being displayed on the list.

Note in the following screenshot how several vendors appear in this list now that the customer filter has been removed:

Vendor	Contact	Title	Location	Email
Azure Interior	Brandon Freeman	Creative Director	Jonesboro, United States	brandon.freeman55@example.com
Azure Interior	Colleen Diaz	Business Executive	Jonesboro, United States	colleen.diaz83@example.com
Deco Addict	Addison Olson	Sales Representative	Franklin, United States	addison.olson26@example.com
Deco Addict	Douglas Fletcher	Functional Consultant	Franklin, United States	douglas.fletcher51@example.com
Deco Addict	Floyd Steward	Analyst	Franklin, United States	floyd.steward34@example.com
Gemini Furniture	Oscar Morgan	Order Clerk	Tampa, United States	oscar.morgan11@example.com
Gemini Furniture	Edwin Hansen	Marketing Manager	Tampa, United States	edwin.hansen58@example.com
Gemini Furniture	Jesse Brown	Senior Consultant	Tampa, United States	jesse.brown74@example.com
Gemini Furniture	Lorraine Douglas	Functional Consultant	Asheville, United States	lorraine.douglas35@example.com
Ready Mat	Billy Fox	Production Supervisor	Columbia, United States	billy.fox45@example.com
Ready Mat	Edith Sanchez	Analyst	Columbia, United States	edith.sanchez68@example.com
Ready Mat	Sandra Neal	Sales Manager	Columbia, United States	sandra.neal80@example.com
Ready Mat	Julie Richards	Financial Manager	Columbia, United States	julie.richards84@example.com
Ready Mat	Kim Snyder	Senior Associate	Columbia, United States	kim.snyder96@example.com

In this instance, if you clear the **Customers** filter by clicking on the close box in the tag, you will have a list with not only customers, but partners, users, suppliers, and contacts as well. The **Customers** filter is applied by default in this view.



Sometimes, users can get confused if they accidentally remove the filter. If you are not getting the results you expect, always double-check the filter in the top-right corner and, if necessary, navigate away from the view and back again to refresh the default filter.

Basic searches are handled easily in Odoo. Just go into the search box, begin typing, and press the *Enter* key. Odoo will then look at the primary search fields for the type of data you are searching for and show you the results in the list or **kanban view**.

In the following screenshot, you can see a simple search:

The screenshot shows the Odoo Sales module interface. At the top, there's a navigation bar with tabs: Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the far right, it shows 'Mitchell Admin'. Below the navigation bar, the main area is titled 'Customers'. There are buttons for 'Create' and 'Import'. The search bar contains 'Name Furniture'. Below the search bar are buttons for 'Filters', 'Group By', 'Favorites', and pagination '1-1 / 1'. The results list a single customer entry: 'Gemini Furniture' with a logo, 'Consulting Services' and 'Vendor / Desk Manufacturers' listed under categories, 'Tampa, United States' as the location, the email 'gemini.furniture39@example.com', and a balance of '\$ 13'.

In this example, Odoo has returned all the customers that have **Furniture** in their name.

Now, we can see that there are two filters applied: the default filter, **Customers**, which was there when we opened the customer list; and the **Name** filter, which will limit those customers to just the names that include **Furniture**.



The small space between the two filter tags means that both conditions are required for a record to be included in the results list (that is, *A and B*).

When two filter tags are butted up against one another without a space between them, it shows that records may meet either condition (that is, *A or B*).

Odoo will remember your search criteria as you move between the list, kanban, and form views. Once you go to another menu item, however, the search criteria will reset to the default search when you return.

As you type in the search box, before hitting the *Enter* key, Odoo will display the available filters in a small drop-down list directly under the search box:

The screenshot shows the Odoo Sales module interface. At the top, there's a navigation bar with tabs: Sales, Orders, To Invoice, Products, Reporting, Configuration, and a user dropdown for Mitchell Admin. Below the navigation bar, the main area is titled "Customers". It features two buttons: "Create" and "Import". There are four cards representing customer records:

- Azure Interior**: Services, Jonesboro, United States, azure.interior24@example.com. It includes a logo for "AZURE - interior -".
- Deco Addict**: Vendor, Franklin, deco.addict. It includes a logo for "DECO ADDICT".
- Lumber Inc**: Asheville, United States, lumber.inv92@example.com. It includes a logo for "LUMBER INC".
- The Jack**: Auburn, jackson. It includes a logo for "Jackson Group".

To the right of the cards, a search bar shows "Customers R". A dropdown menu lists various search and filter options starting with "Search Name for: R".

- Search Name for: R
- Search Related Company for: R
- Search Tag for: R
- Search Salesperson for: R
- Search Phone for: R
- Filter on: My Contacts
- Filter on: Individuals
- Filter on: Customers
- Filter on: Vendors
- Filter on: Archived
- Filter on: Late Activities
- Filter on: Future Activities
- Group by: Salesperson
- Group by: Company
- Group by: Country

Type the letter R into the search box; you will notice how the R is now in bold where it will be applied to the filter. Additionally, note that to the left of **Search Related Company for: R** there is a small triangle. Clicking on this triangle will expand the results in the list:

The screenshot shows the Odoo Sales module's customer list view. At the top, there is a search bar with the placeholder "Customers" and a search icon. A dropdown menu is open, showing search suggestions starting with "R". The suggestions include "Search Name for: R", "Search Related Company for: R", and a expanded list under "Search Related Company for: R" which includes "Azure Interior", "Deco Add", "Gemini Furniture", "Lumber Inc", "Ready Mat", "The Jackson Group", "Wood Corner", and "YourCompany". Other suggestions listed are "Search Tag for: R", "Search Salesperson for: R", "Search Phone for: R", and various filters like "Filter on: My Contacts", "Filter on: Individuals", etc.

Customer	Description	Address	Email	Phone
Azure Interior	■ Services	Jonesboro, United States	azure.Interior24@example.com	\$ 1
Lumber Inc		Asheville, United States	lumber-inv92@example.com	\$ 1
Deco Add	■ Vendor	Franklin, deco.addict		\$ 2
The Jack		Auburn, jackson		

In this way, you may navigate directly to a specific customer quickly and without ever having to leave the search box. The more letters you type into the search box, the shorter the list of results will become, so you may quickly narrow down your search.

Using filters in the list views

Odoo provides default filters for all the list views. Applying a filter will limit the records that Odoo is displaying. You can apply one or more filters depending on your needs. The available filters will vary depending on the data that you are viewing.

For example, the **Products** view will have a completely different set of filters and group options than the **Customers** view:

Acoustic Bloc Screens [FURN_6666] Price: \$ 2,950.00	Cabinet with Doors [E-COM11] Price: \$ 14.00	Chair floor protection Price: \$ 12.00
Conference Chair 2 Variants Price: \$ 16.50	Corner Desk Black [FURN_1118] Price: \$ 85.00	Corner Desk Right Sit [E-COM08] Price: \$ 147.00
Customizable Desk 6 Variants Price: \$ 750.00	Deposit [Deposit] Price: \$ 150.00	Desk Combination [FURN_7800] Price: \$ 450.00
Desk Stand with Screen [FURN_7888] Price: \$ 2,100.00	Drawer [FURN_8855] Price: \$ 3,645.00	Drawer Black [FURN_8900] Price: \$ 25.00
Flipover [FURN_9001] Price: \$ 1,950.00	Four Person Desk [FURN_8220] Price: \$ 23,500.00	Hotel Accommodation [EXP_HA] Price: \$ 400.00
Individual Workplace [FURN_0786] Price: \$ 885.00	Large Cabinet [E-COM07] Price: \$ 320.00	Large Desk [E-COM09] Price: \$ 1,799.00

While each search box will have different default filters and **Group By** options, the functionality will be the same. In Odoo 12, you can access all the search features by clicking on the small magnifying glass with the plus sign in it to the right of the search box:



When you click on this, Odoo will expand the search area to show all the search features. Let's return to the product list and click on the magnifying glass to bring up the advanced search options:

In Odoo 12, the advanced search tools have been consolidated into **Filters**, **Group By**, and **Favorites**. Let's first explore the data using **Filters** to limit the records that are displayed:

The screenshot shows the Odoo Sales module interface. At the top, there are tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the far right, it shows 'Mitchell Admin'. Below the tabs, there's a 'Products' section with 'Create' and 'Import' buttons. The main area displays a grid of product cards. Each card includes an icon, the product name, a description, and a price. To the right of the grid is a sidebar with a 'Filters' dropdown set to 'Can be Sold'. Other filter options shown are 'Can be Purchased' and 'Archived'. There are also buttons for 'Group By', 'Favorites', and a search bar. The page number '1-29 / 29' is at the bottom.

In the example drop-down list in the preceding screenshot, you can see the check mark that has been applied next to the **Can be Sold** filter. The tags for the currently selected filter are also displayed inside the search box. Clicking on a filter applies the filter immediately and refreshes the results list.

Naturally, the list of available filters will change depending on which set of records you are viewing. You can add multiple filters, and Odoo will return the records that match all the filters you have applied. Clicking on a filter that is already checked will remove that filter.

Grouping information

In addition to filtering your results, you can also group data in most kanban and list views using the **Group By** option. When you group data in a kanban view, you will get a column for each category. You can then use the horizontal scroll bar at the bottom of your window to look through the items.

This will be ineffective for items with a very large number of groups:

The screenshot shows the Odoo Sales module interface. At the top, there's a navigation bar with tabs: Sales, Orders, To Invoice, Products, Reporting, Configuration, and a user dropdown for 'Mitchell Admin'. Below the navigation is a search bar with filters for 'Customers' and 'Country', and a search input field. To the right of the search bar are buttons for 'Filters', 'Group By', 'Favorites', and various view modes.

The main area displays a kanban view of customers. On the left, there's a group header for 'France' containing one customer card for 'Deco Addict'. On the right, there's a group header for 'United States' containing five customer cards: 'Azure Interior', 'Gemini Furniture', 'Lumber Inc', 'The Jackson Group', and a card for 'YourCompany, Joel Willis' which includes a small profile picture of a man.

A context menu is open over the 'YourCompany, Joel Willis' card. The menu has a 'Country' option checked, and other options include 'Salesperson', 'Company', and 'Add Custom Group'. The menu also has a 'Group By' button at the bottom.

Customer	Country	Description
Deco Addict	France	Vendor / Desk Manufacturers Franklin, France deco.addict82@example.com \$ 2
Azure Interior	United States	Services Jonesboro, United States azure.Interior24@example.com
Gemini Furniture	United States	Consulting Services Vendor / Desk Manufacturers Tampa, United States gemini.furniture39@example.com \$ 13
Lumber Inc	United States	Asheville, United States lumber-inv92@example.com \$ 1
The Jackson Group	United States	Auburn, United States jackson.group82@example.com
YourCompany, Joel Willis	United States	Bayonne, United States joel.willis63@example.com \$ 2

The preceding screenshot shows **Customers** in a kanban view, demonstrating how a user will need to scroll not only up and down, but also left and right to get a view of all the items. We have changed the country of one of the customers to **France** to show how the kanban view displays groups.

Grouped data is often more easily represented in a list view. When you group data in a list, a little triangle appears to the left of each group header.

Clicking on this triangle will display the rows grouped under that header, as shown in the following screenshot:

The screenshot shows the Odoo Sales module interface. At the top, there are tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the right, there are user icons and a dropdown for 'Mitchell Admin'. Below the tabs, a search bar has 'Customers' and 'Country' selected, with a 'Search...' button. Underneath the search bar are buttons for 'Filters', 'Group By', 'Favorites', and view modes. The main area is titled 'Customers' and shows a list of companies. A 'Create' button is at the top left of the list. The list includes columns for 'Name', 'Phone', and 'Email'. It shows two groups: 'France (1)' containing one entry ('Deco Addict'), and 'United States (5)' containing five entries ('Azure Interior', 'Gemini Furniture', 'Lumber Inc', 'The Jackson Group', and 'YourCompany, Joel Willis'). Each entry has a checkbox next to it and its contact information.

We have filtered our list by customers that are companies by selecting the **Companies** option in the **Filters** menu. We also grouped our data by category by selecting **Country** in the **Group By** menu. Next, we expanded the **France(1)** country section by clicking on the small triangle to the left. You can then see the list of companies that are in France. As with filters, clicking on **Companies** again will remove the grouping. You can also nest groups inside of other groups by simply selecting additional items under **Group By**.



Grouping can be a great way to look at data. Unfortunately, with extremely large datasets, grouping lists can be very slow because far more records must be processed if you are filtering and browsing data.

Performing a custom search

While the default filters may help us to find most of the data records we seek, it is inevitable that there will come a time when we will need a more customized search. To create a custom search, open up the **Filters** drop-down menu, and then click on **Add Custom Filter** to expand the available options. Here, we will get a drop-down list of fields that can be used to set our search criteria:

The screenshot shows the Odoo web interface for managing customers. On the left, a list of customers is displayed with columns for Name and Phone. A context menu is open over a record, listing various fields like Account Payable, Account Receivable, Active, etc. Below the menu, a search bar is set to 'Account Payable' with the operator 'contains'. At the bottom right of the search bar are 'Apply' and 'Add a condition' buttons.

Choosing a field from the list will allow you to select a criterion from the available search operators and specify the data for which you wish to search. Click on **Add a condition** to enter further criteria. Clicking on the small **x** icon to the right of a search condition will delete that condition from your custom filter.

In Odoo, you will often find it a best practice to make records inactive when they are no longer required. For example, if you discontinue a product, you will often find yourself unable to delete that product because there are transactions tied to it. Therefore, you will wish to deactivate that product record. By default, Odoo will hide inactive records. If you need to retrieve inactive records, use a custom filter to create a condition where the **Active** field is false, and then apply it to this filter.



You can continue to add additional criteria to your custom filter. When you have specified all the criteria you wish to use in your search, click on the **Apply** button to apply the custom filter:

The screenshot shows the Odoo Sales module interface. On the left, there's a grid of product cards. On the right, a sidebar displays a list of products and a custom filter configuration. The custom filter is set to 'Active' and 'is true'. It also includes a condition for 'Sale Price' being 'greater than or equal to' 100. The 'Apply' button is visible at the bottom of the filter sidebar.

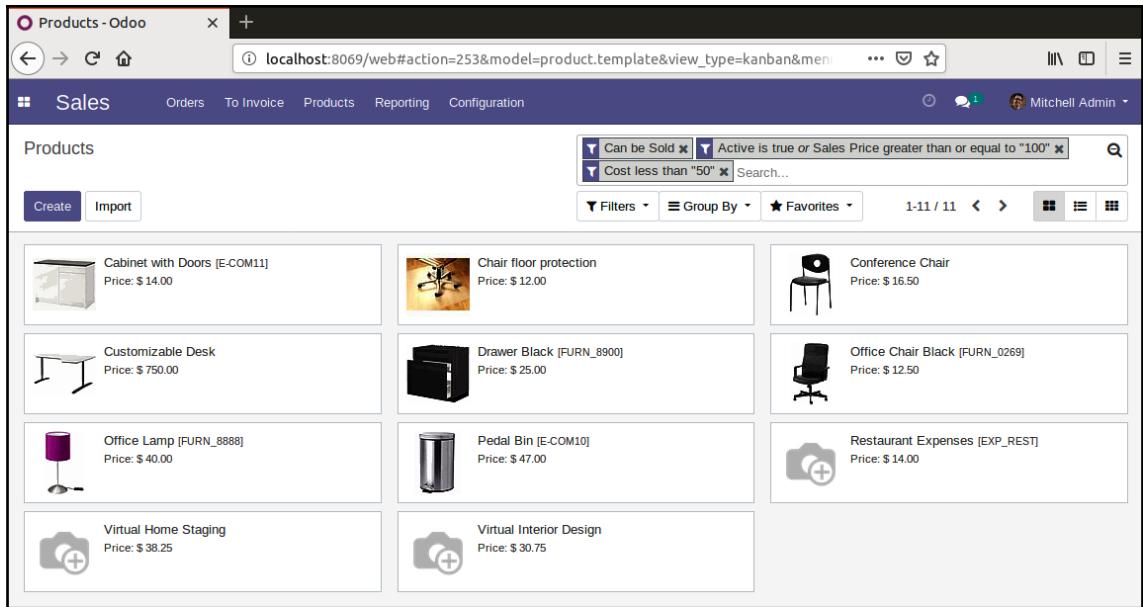
Product	Description	Price
Acoustic Bloc Screens [FURN_6668]	Price: \$ 2,950.00	
Cabinet with	Price: \$ 14.00	
Conference Chair	2 Variants Price: \$ 16.50	
Corner Desk	Price: \$ 85.00	
Customizable Desk	6 Variants Price: \$ 750.00	
Deposit [Deposit]	Price: \$ 150.00	
Desk Stand with Screen [FURN_7888]	Price: \$ 2,100.00	
Drawer [FURN_	Price: \$ 3,645.00	
Flipover [FURN_9001]	Price: \$ 1,950.00	
Four Person	Price: \$ 23,500.00	
Individual Workplace [FURN_0789]	Price: \$ 885.00	
Large Cabinet	Price: \$ 320.00	
Chair floor protection	Price: \$ 12.00	
Corner Desk Right Sit [E-COM08]	Price: \$ 147.00	
Desk Combination [FURN_7800]	Price: \$ 450.00	
Drawer Black [FURN_8900]	Price: \$ 25.00	
Hotel Accommodation [EXP_HA]	Price: \$ 400.00	
Large Desk [E-COM09]	Price: \$ 1,799.00	

In the custom filter option, we have specified two conditions: **Active** must be **is true**, or the **Sale Price** for the customer must be **greater than or equal to** a value of **100**. Many users can get confused and believe that this filter would imply that both **Active** and **Sale Price** must meet our criteria, but that is not the case.

Specifying multiple advanced searches

As you saw in our previous example, Odoo will always use an **or** operation between each of the conditions you add to the search. But what if we wanted to have a search where the record is **Active** and the **Sale Price** is **greater than or equal to 100**? To accomplish this, you must first apply the custom filter with only the **Active** condition defined. That will limit the results to only active products. Then, you can go back and add a second custom filter that only contains the **Sale Price** is **greater than or equal to 100** condition.

Just remember that if you want both conditions to be true, they must be applied separately. If you want either of the conditions to be true, then add them together in one search:



The screenshot shows the Odoo web interface for the 'Products' module in Kanban view. The URL is `localhost:8069/web#action=253&model=product.template&view_type=kanban&menu_id=100`. The top navigation bar includes 'Sales', 'Orders', 'To Invoice', 'Products', 'Reporting', and 'Configuration'. A user 'Mitchell Admin' is logged in. The main area displays a grid of product cards. At the top of the grid, there are three filters: 'Can be Sold' (unchecked), 'Active is true or Sales Price greater than or equal to "100"' (checked), and 'Cost less than "50"' (unchecked). Below the filters are buttons for 'Filters', 'Group By', 'Favorites', and page navigation (1-11 / 11). The products listed are:

Image	Name	Code	Price
	Cabinet with Doors [E-COM11]		Price: \$14.00
	Chair floor protection		Price: \$12.00
	Conference Chair		Price: \$16.50
	Customizable Desk		Price: \$750.00
	Drawer Black [FURN_8900]		Price: \$25.00
	Office Chair Black [FURN_0269]		Price: \$12.50
	Office Lamp [FURN_8888]		Price: \$40.00
	Pedal Bin [E-COM10]		Price: \$47.00
	Restaurant Expenses [EXP_REST]		Price: \$14.00
	Virtual Home Staging		Price: \$38.25
	Virtual Interior Design		Price: \$30.75

In the preceding screenshot, we created a search that will return **Products** that can be sold and are **Active** or have a **Sale Price greater than or equal to \$100**, and the cost of the product must be less than \$50.

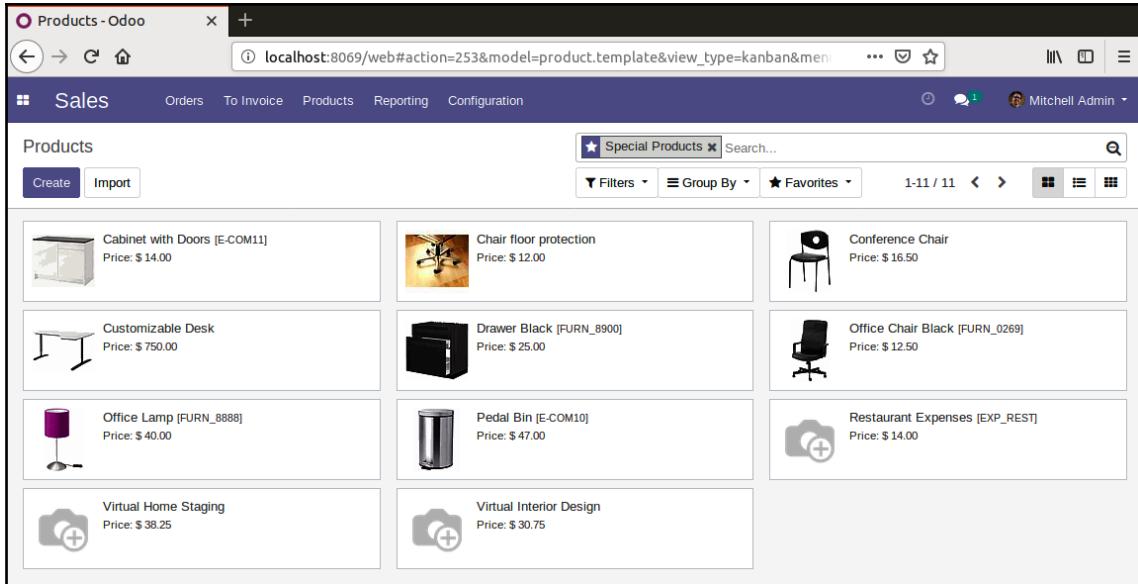
Saving your favorite filters

While advanced searches are quite powerful, they can often take a bit of time to configure to get the results in the manner in which you want them. Fortunately, Odoo allows you to save your searches so that you will not have to build them from scratch each time. To save a custom search, click on the **Favorites** drop-down list (which is the little triangle next to **Save current search**), provide a name for the search, and then click on **Save**:

The screenshot shows the Odoo Sales Kanban view for Products. At the top, there are filters: 'Can be Sold' (selected), 'Active is true or Sales Price greater than or equal to "100"' (selected), and 'Cost less than "50"' (selected). Below the filters is a search bar. To the right of the search bar is a 'Favorites' dropdown menu. The menu is open, showing the option 'Save current search' with the input field 'Special|Products'. There are also checkboxes for 'Use by default' and 'Share with all users'. A 'Save' button is at the bottom of the menu. The main area displays a grid of products:

Image	Name	Code	Price
	Cabinet with Doors [E-COM11]		Price: \$14.00
	Chair floor protection		Price: \$12.00
	Customizable Desk		Price: \$750.00
	Drawer Black [FURN_8900]		Price: \$25.00
	Office Lamp [FURN_8888]		Price: \$40.00
	Pedal Bin [E-COM10]		Price: \$47.00
	Virtual Home Staging		Price: \$38.25
	Virtual Interior Design		Price: \$30.75

Once you click on **Save**, the filter is added to your list of **Favorites** and can be applied, just like the default Odoo filters. In addition to this, you also have the option to save the custom filter for all users, and even set a custom filter as the default filter to be applied when you bring up the list:



The screenshot shows the Odoo Sales Kanban view. At the top, there's a search bar with the text 'Special Products' and a 'Filters' dropdown. Below the search bar, there are buttons for 'Create' and 'Import'. The main area displays a grid of products. Each product card includes an image, the product name, and its price. The products shown are:

- Cabinet with Doors [E-COM11] - Price: \$14.00
- Chair floor protection - Price: \$12.00
- Conference Chair - Price: \$16.50
- Customizable Desk - Price: \$750.00
- Drawer Black [FURN_8900] - Price: \$25.00
- Office Chair Black [FURN_0269] - Price: \$12.50
- Office Lamp [FURN_8888] - Price: \$40.00
- Pedal Bin [E-COM10] - Price: \$47.00
- Restaurant Expenses [EXP_REST] - Price: \$14.00
- Virtual Home Staging - Price: \$38.25
- Virtual Interior Design - Price: \$30.75

In the preceding screenshot, we applied the custom filter that we just saved: **Special Products**. As you can see, the criteria at the top no longer shows all the detail in the advanced search, and instead uses the name you provided when you saved the custom filter.

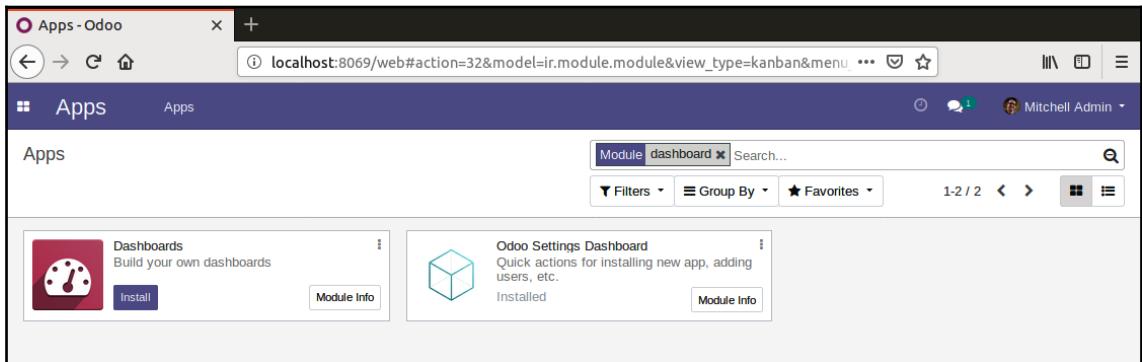


There is no easy way for an end user to see what the criteria of their search is after they have named and saved their search for later use. Just like in our example, products that are **Special** are all we will see when returning to the search later. This is extremely vague. Until Odoo provides an easier method, users should be encouraged to document their searches well and use clearly worded names for their custom filters.

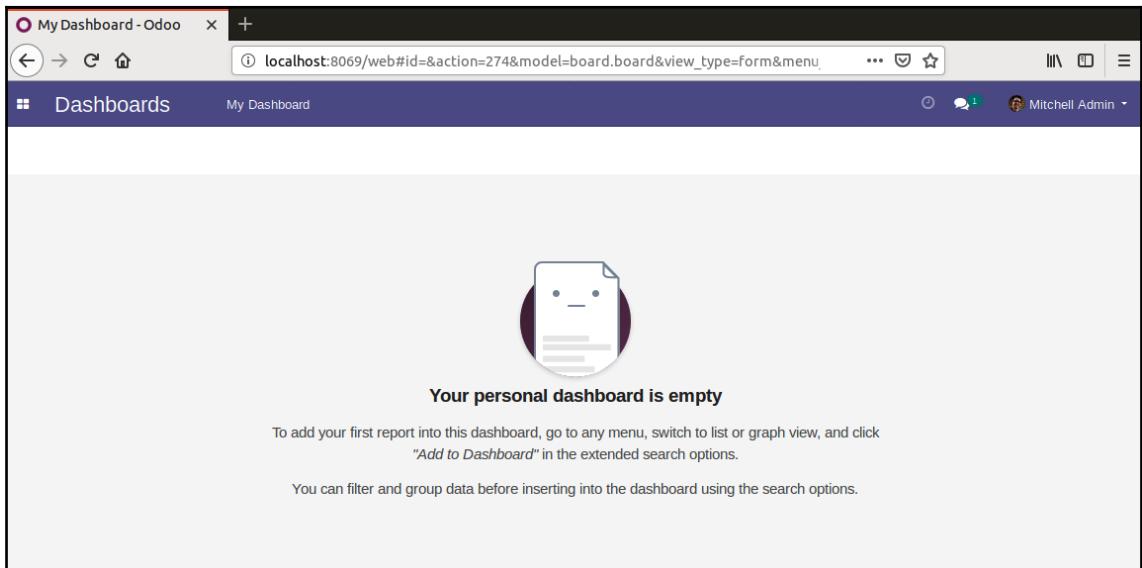
The ability to save advanced searches into your own custom filters and make them available for other users allows you to better customize Odoo for your business requirements.

Creating custom dashboards in Odoo 12

Dashboards allow you to take information that you need to look at frequently and put it together in one place. In previous versions of Odoo, the installation came with dashboards, even if you never used them. In Odoo 12, you can add custom dashboard support by installing the **Dashboards** module:



Odoo has a very flexible dashboard system. Each user has a personal dashboard named **My Dashboard** that is provided when you install the **Dashboards** module. Initially, this dashboard will be empty, except for some useful instructions on customizing your dashboard:



Let's examine how we can add our custom filter from the previous section to our dashboard. To do this, return to the products, choose the list view, and apply the custom filter:

The screenshot shows the Odoo web interface for the 'Products' module. The top navigation bar includes links for Sales, Orders, To Invoice, Products, Reporting, and Configuration, along with a user dropdown for 'Mitchell Admin'. Below the header is a search bar with placeholder text 'Special Products' and a magnifying glass icon. To the right of the search bar are buttons for 'Filters', 'Group By', and 'Favorites'. A status bar at the bottom indicates '1-11 / 11' with navigation arrows. The main content area displays a table of products with columns for Internal Reference, Name, Sales Price, Cost, Product Category, and Product Type. The table lists various items like 'Cabinet with Doors', 'Chair floor protection', and 'Customizable Desk', each with its respective details.

To add a new result set to your dashboard, open up the **Favorites** drop-down menu and simply click on the little triangle next to **Add to my Dashboard**. By default, Odoo will prompt you to add the search list to your own personal dashboard. However, if you wish, you can add the results to any dashboard by selecting the name of that dashboard in the list and clicking on the **Add** button:

This screenshot is similar to the previous one but includes a context menu. The 'Special Products' dropdown has been expanded, revealing options: 'Save current search', 'Add to my Dashboard', and a text input field containing 'Speical Products'. Below this, a large 'Add' button is visible. The rest of the interface and product list are identical to the first screenshot.

In this example, the current **Special Products** filter we created in the previous step will be added to **My Dashboard**, which can be found as the first option under the **Dashboards** menu.



Odoo dashboards will use whatever view of the data (such as list, kanban, calendar, pivot table, bar graph, line graph, or pie chart) you are currently using when you save the new item to your dashboard. A dashboard with a variety of different views may seem more visually appealing, but you'll get the most use out of your dashboard if you select a view that best summarizes the dataset.

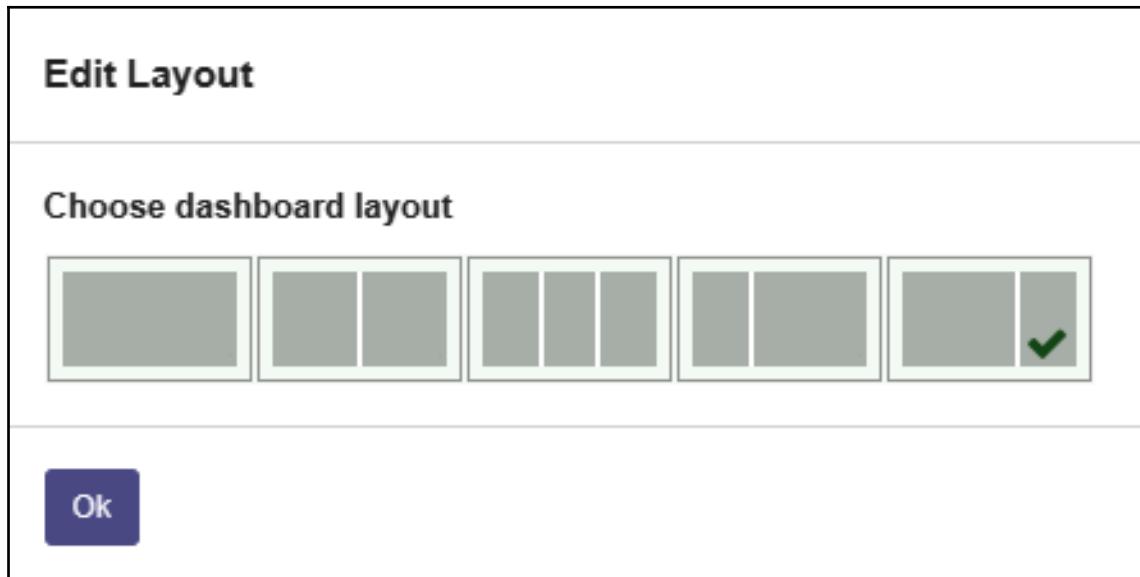
The build of Odoo 12 that we were running required that you *Shift + refresh* your browser for any new additions to the dashboard to show up. Here we see our updated **My Dashboard** screen, which now contains a table of data we created from our custom filter of **Special Products**:

Internal Reference	Name	Sales Price	Cost	Product Category	Product Type
E-COM11	Cabinet with Doors	14.00	12.50	All / Saleable / Office Furniture	Consumable
	Chair floor protection	12.00	0.00	All / Saleable / Office Furniture	Consumable
	Conference Chair	16.50	0.00	All / Saleable / Office Furniture	Consumable
	Customizable Desk	750.00	0.00	All / Saleable / Office Furniture	Consumable
FURN_8900	Drawer Black	25.00	20.00	All / Saleable / Office Furniture	Consumable
FURN_0269	Office Chair Black	12.50	18.00	All / Saleable / Office Furniture	Consumable
FURN_8888	Office Lamp	40.00	35.00	All / Saleable / Office Furniture	Consumable
E-COM10	Pedal Bin	47.00	10.00	All / Saleable / Office Furniture	Consumable
EXP_REST	Restaurant Expenses	14.00	8.00	All / Expenses	Service
	Virtual Home Staging	38.25	25.50	All / Saleable / Services	Service
	Virtual Interior Design	30.75	20.50	All / Saleable / Services	Service

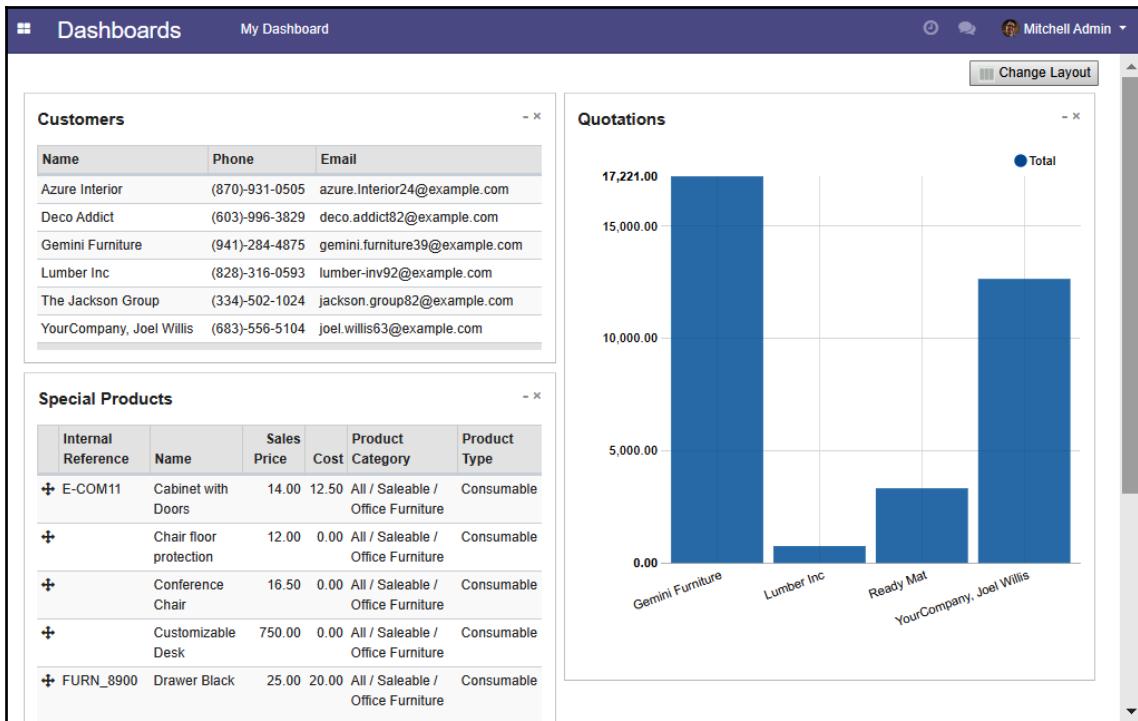
The title of your dashboard report area does not need to be named the same as the custom filter it is built upon. You may have already noticed that the word **Special** was misspelled for the dashboard, but not for the filter. This clearly demonstrates that the title of the dashboard item can be named differently from the filter. Additionally, it is not possible to rename your custom filters or your dashboard item titles without recreating them.

Odoo provides a variety of layouts so that you can customize the appearance of the dashboard according to your preference. For example, you may wish to have two columns of lists summarizing your sales; alternatively, if there are view columns, you may choose to have a column of three lists.

Clicking on the **Change Layout** button will bring up a small pop-up window to allow you to select an alternative layout as shown below:



In the top-right corner of each item added to the dashboard, you can click on the little underscore icon to collapse the report area down to just its title. To arrange an item on your dashboard, simply click and drag the item to drop it in the desired location. Finally, you can remove an item from the dashboard by clicking on the close box in the top-right corner of the item. The following screenshot displays the dashboards with a graphical representation:

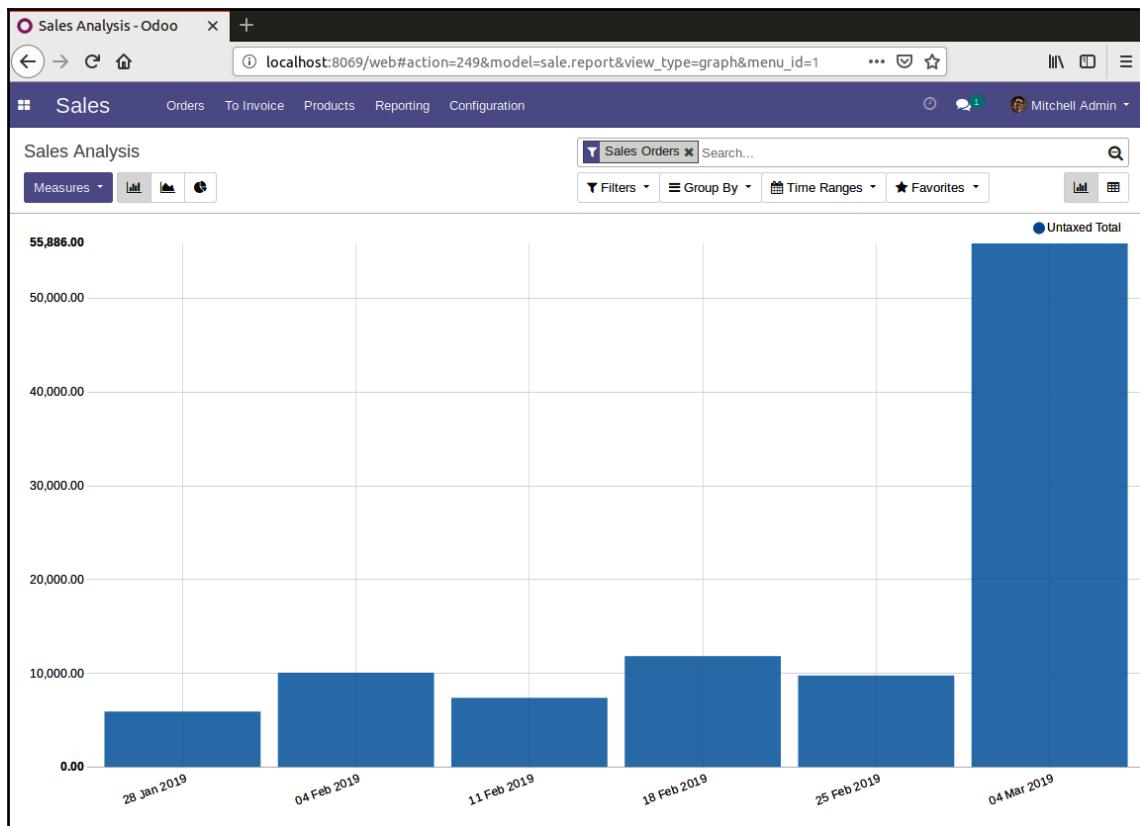


In this example, we have added a few more items to the dashboard and arranged them into two columns. Adding a graph is just as easy as adding a list view to the dashboard. In this example, we went to **Quotations**, changed the view to **Graph**, and then added it to our dashboard.

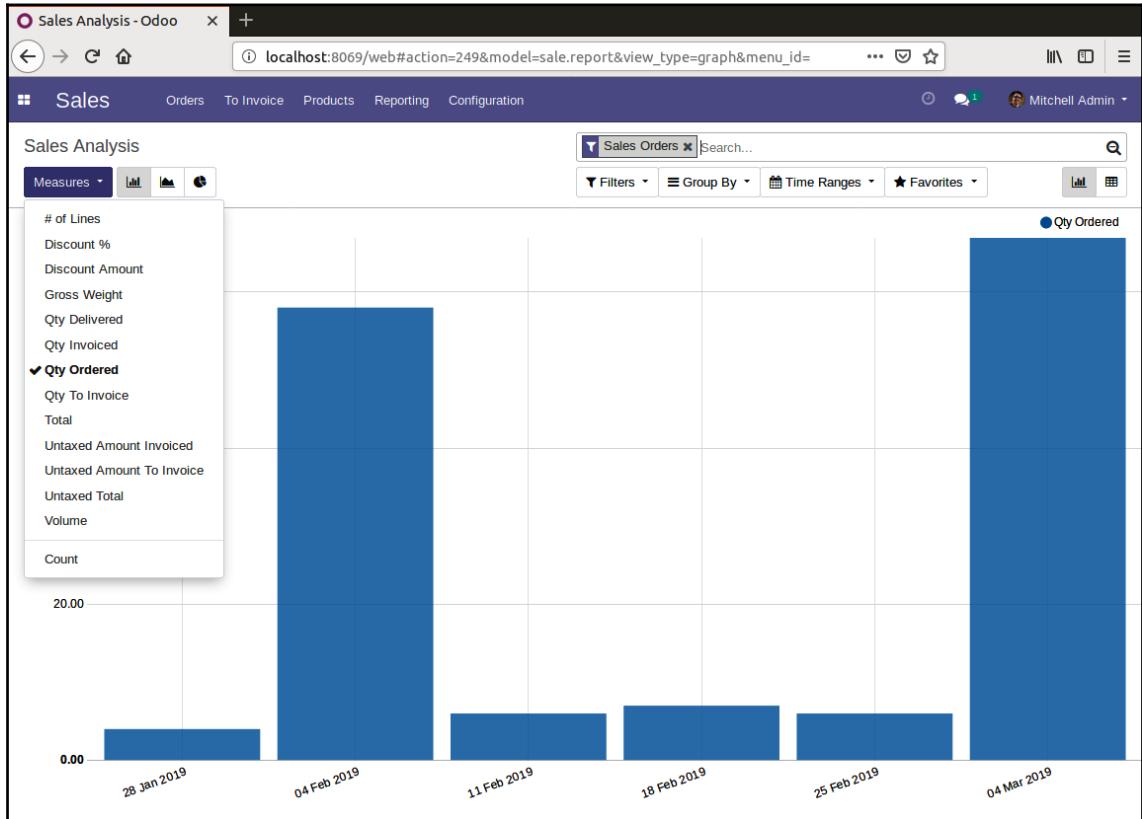
Looking at Odoo's business intelligence features

In addition to powerful search features and dashboards, Odoo also provides a robust business intelligence framework that allows real-time reporting from most applications. You can see an example of Odoo's business intelligence features by opening the **Sales** application and clicking on **Sales** under the **Reporting** heading.

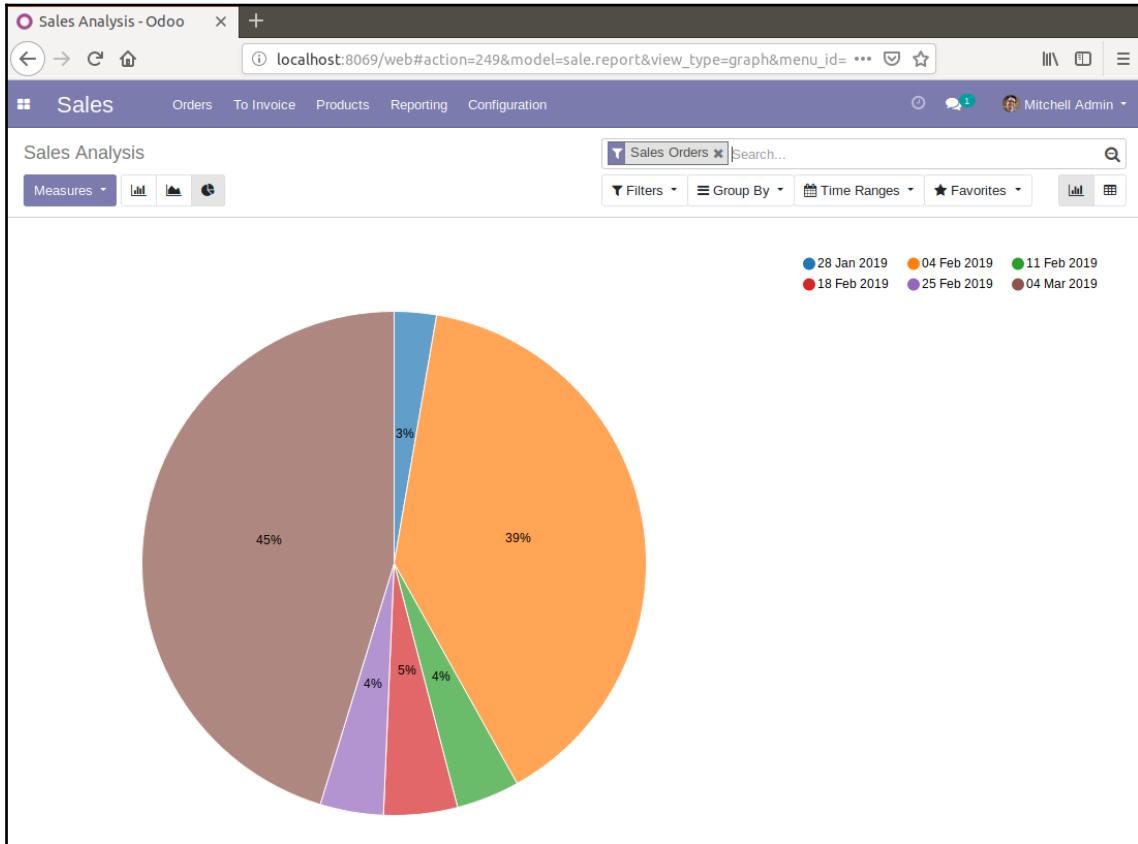
As you can see, by default, Odoo has provided a bar graph of our sales by month. We can then go under **Measures** to change the data, and the graph displays the following:



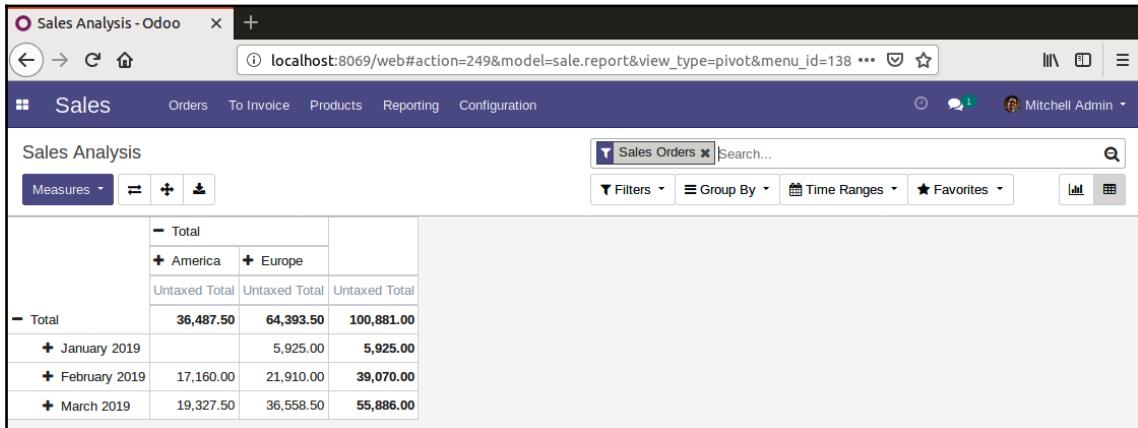
Here, we are currently viewing the **Untaxed Total** measure, but you can change this to any of the other measures for your graph simply by selecting from the list. Let's change the graph to display **Qty Ordered** instead:



You can see that the description on the right-hand side has changed to show the measure that we have selected. Additionally, note that you can still apply **Filters** and save **Favorites** as well. Finally, if you wish, you can look at the data as a line graph or a pie chart using the icons next to the **Measures** drop down. In the following screenshot, we have changed the graph into a pie chart:



The available **Measures** will, of course, change depending on the application and specific report you are working with. In addition to this, you can also look at the data in a grid view instead of a graph. For many reports, this is a more practical way of viewing the data. Here, we have clicked on the icon in the top-right corner of the view in order to display the data in a grid instead of a graph:



The screenshot shows the Odoo Sales Analysis interface. At the top, there's a header bar with a back/forward button, a search bar containing 'localhost:8069/web#action=249&model=sale.report&view_type=pivot&menu_id=138 ***', and various icons. Below the header is a navigation menu with 'Sales' selected, followed by 'Orders', 'To Invoice', 'Products', 'Reporting', and 'Configuration'. On the right side of the header, there's a user profile for 'Mitchell Admin'.

The main area is titled 'Sales Analysis' and contains a 'Sales Orders' search bar with a magnifying glass icon. Below the search bar are several filter buttons: 'Filters', 'Group By', 'Time Ranges', and 'Favorites'. There are also download icons for Excel and CSV.

The central part of the screen is a pivot table grid. The columns are labeled 'Untaxed Total', 'Untaxed Total', and 'Untaxed Total'. The rows show data for 'Total', 'America', 'Europe', and months from January 2019 to March 2019. The data values are: Total (36,487.50, 64,393.50, 100,881.00), America (5,925.00, 5,925.00), Europe (17,160.00, 21,910.00, 39,070.00), January 2019 (19,327.50, 36,558.50, 55,886.00).

As you can see, there are also + and - controls in the rows and columns. These allow you to add and remove data items in the grid. When showing the data grid, you also have access to a small download button to export your data into an Excel file.

Summary

In this chapter, we examined Odoo's advanced searches and dashboards. Advanced searching allows you to search based on a variety of fields and to save your searches so that you can easily pull them up later. Using these features, you can find the data you are looking for and place data that you frequently need into your own personal dashboard more easily.

In the next chapter, we will learn about Odoo's **Website Builder** and compare it to other content management systems. While building our website, we'll experiment with different page layouts, content blocks, menus, and themes. Finally, we'll explore Odoo's features designed to implement best practices for **search engine optimization (SEO)**.

11

Building a Website with Odoo

In this chapter, we will look at perhaps the most important business application that was added to Odoo in version 8: the Odoo Website Builder application. The Website Builder allows you to quickly create not just a home page for your company, but an entire website complete with multiple pages, a menu system, a contact form, themes, and more.

Topics covered in this chapter include the following:

- A brief introduction to **content management systems (CMsEs)** and how they make it possible to manage websites
- Modifying pages with Odoo's **Website Builder**
- Inserting and customizing blocks
- Important Odoo website blocks and how to use them
- Editing the menu of your website and organizing pages
- Selecting themes for your website
- Promoting your website

What is a CMS?

The **Website Builder** application available for Odoo 12 is commonly known as a CMS. A CMS is a collection of tools that allows you to structure, organize, and manipulate your website without having to interact directly with the inner workings of your website. A key feature of a CMS is the ability for non-programmers and those with little technical expertise to create and edit content on the website once the initial structure of the site has been designed.

In many ways, Odoo is coming into a very crowded market that has a great variety of both open source and paid CMsEs with which you can choose to build your website. Here are a few popular website CMsEs, all of which, at this point, have considerably more configuration options and significantly more installs than Odoo.

WordPress

WordPress is arguably the most popular CMS that companies choose for deploying their website. More than a decade of maturity and a massive install base means there are plenty of themes, add-ons, and professionals that can support a WordPress website. In addition, WordPress is open source and continues to be developed aggressively, and, in more recent versions, is targeting improved social networking features.

Joomla

Joomla is also enjoying great popularity in the crowded CMS market and is written in PHP and is open source as well. This CMS, while perhaps not as often deployed as WordPress, has thousands of available plugins and can be found under some very prominent sites on the internet. A few of the more high-profile sites that use Joomla for their CMS are Harvard University and the Guggenheim Museum.

Drupal

No list of popular CMS solutions would be complete without including **Drupal**. Like the other two, this CMS is also PHP-based and open source. For the most part, Drupal has more advanced capabilities and would be considered for more complex sites than perhaps those you would build in WordPress. There are fewer available themes for Drupal than for Joomla and WordPress, but which has not kept Drupal from being the CMS for very prominent websites, including Popular Science and Sony Music.

Evoq or DotNetNuke

Evoq was previously known as **DotNetNuke (DNN)** and has recently been changed through its own rebranding effort, much like the way OpenERP became Odoo. So, according to their own news release, DNN is no more and is now to be known as Evoq. While not nearly as popular as the other three listed, Evoq has the distinction of being a Windows server-based solution that uses Microsoft's .NET platform. Some big names using Evoq for their CMS are Hilton and Samsung.

Why use Odoo Website Builder as your CMS?

With so many CMS solutions available that have far better support and more mature features, a very valid question is this: why would someone use the Odoo Website Builder as their CMS? Not only is this a good question to ask, it is vital when building a website for your company that you pick the tools that work best for your given situation and requirements. So let's quickly look at some of the pros and cons of using Odoo as the CMS to build your website.

Potential advantages of using Odoo as a CMS

While Odoo is still new and does not have the same kind of proven track record and successful websites as the CMSes that have been previously mentioned, there are still some very compelling reasons to consider Odoo as your CMS:

- It has a one-click setup if you already have Odoo installed.
- It integrates seamlessly with Odoo to leverage many of the applications that are already available for Odoo. This is especially true for the e-commerce application that will be covered in the next chapter.
- It has very easy-to-use features, such as fast page editing and simple controls.
- It has great support for mobile devices.
- A growing number of professional themes will make the Odoo Website Builder an attractive option in the years to come for those who are already using Odoo.
- It has powerful built-in language translation support.
- It has good built-in promotional tools.

Current limitations of using Odoo as a CMS

Despite a growing list of positive reasons to consider Odoo, there are also some reasons why it may not be the CMS for every solution:

- It has very limited support among hosting companies, website designers, and consultants. If your Odoo website breaks, you are reliant on Odoo experts to fix your website.
- The set of themes available that work directly with Odoo is very limited, and the professional themes that are available are expensive.
- It has limited CMS functionality for version control of your web pages.
- The immaturity of the CMS itself may cause volatility in the years ahead as new features are added, making it challenging to move your website between databases of various Odoo versions.
- It has complex and confusing security in its web pages and in the assignment of access permissions to content.
- There is no easy way to move websites or web pages between Odoo databases, which can add to integration, testing, and deployment challenges.

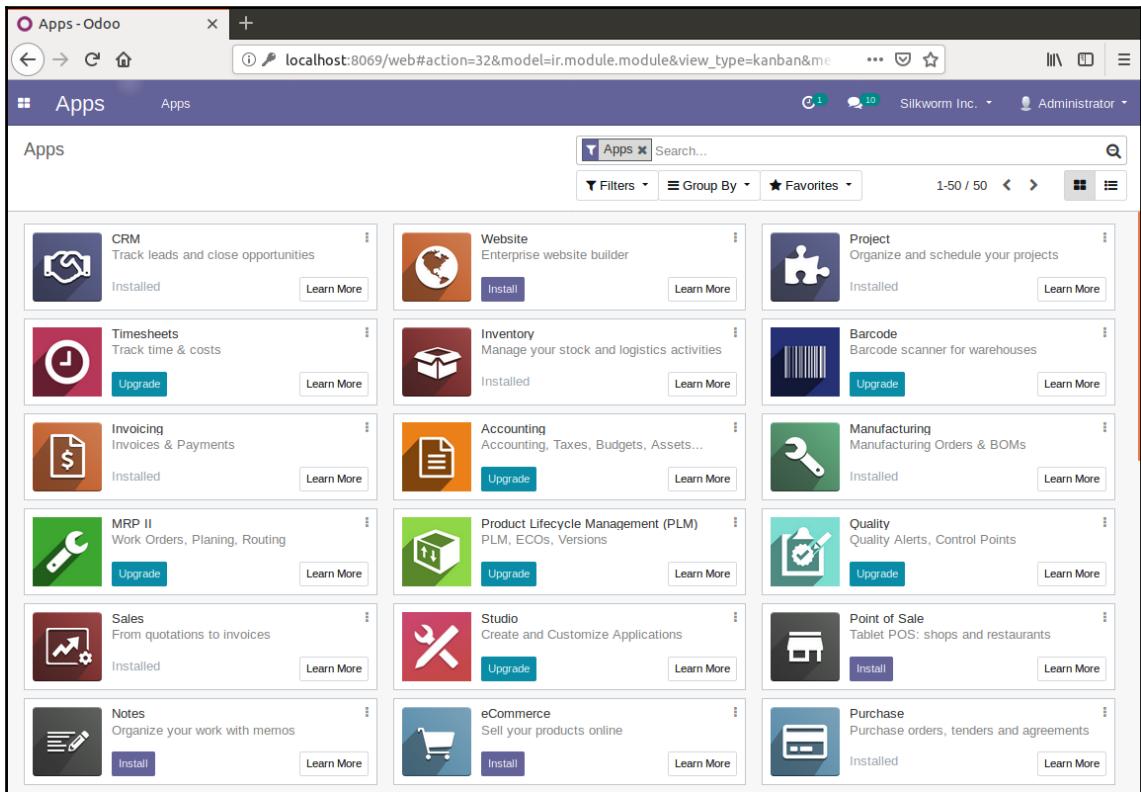
Backing up the website you make in your Odoo database

If you do use Odoo's Website Builder for your website, as has been stated in many of the other chapters, back up your database often. All of the web pages you create are stored inside your database, so you must back it up to make sure you have a copy of your website. Additionally, you will want to make sure you keep your Odoo application backed up as well, because the static themes, images, and CSS files that are located there must be available in order to properly display your website.

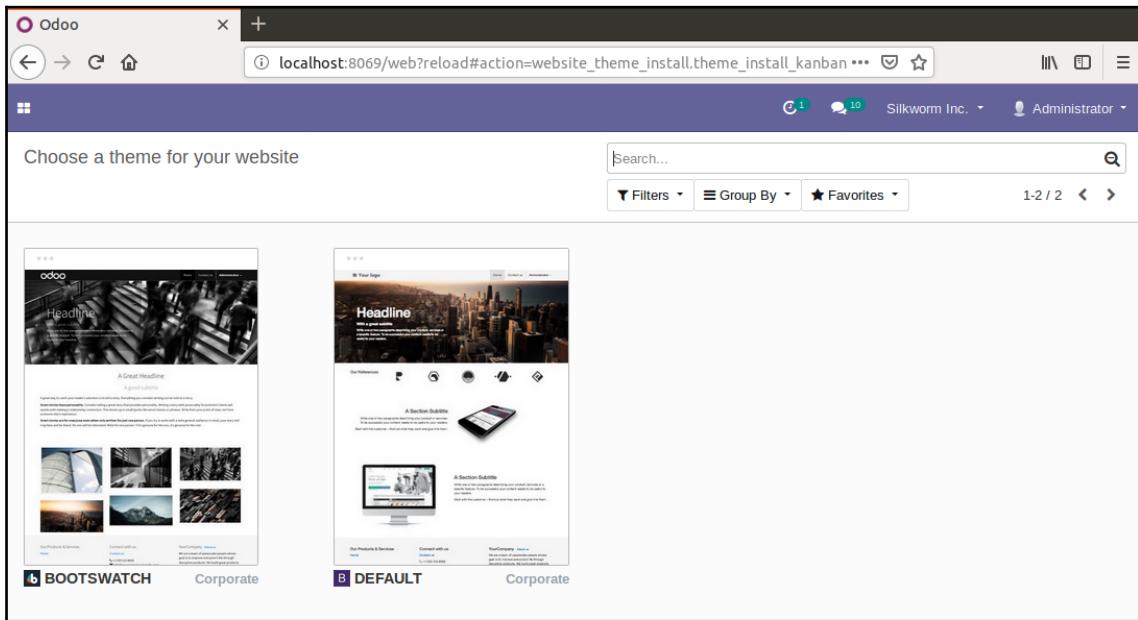
In Chapter 13, *Customizing Odoo for Your Business*, you can find simple instructions on how to back up your database.

Installing the Odoo Website Builder

One of the greatest reasons to consider Odoo's Website Builder is that you can try it out in a matter of seconds. Just install the **Website** application as you would any other Odoo application. Go to **Apps** and search for **Website**:



Once you click **Install**, Odoo will install the required modules and take you to a new screen that will allow you to choose the basic theme for your website:



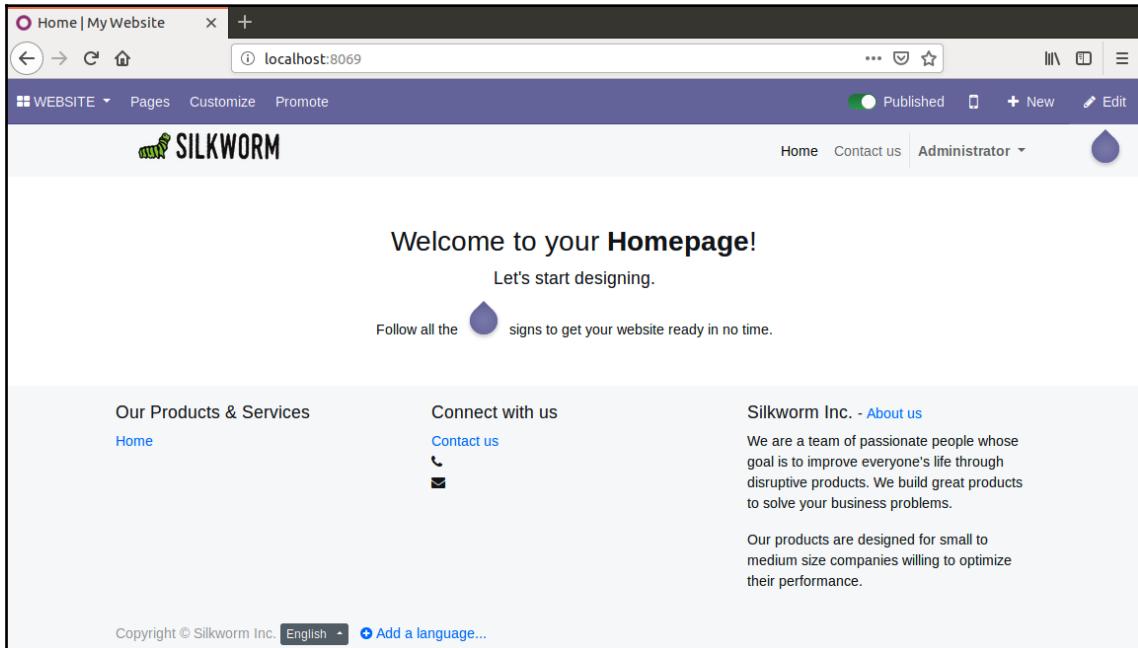
In this build of Odoo 12, you can choose between **BOOTSWATCH** themes and **DEFAULT**. You will also now be provided with a thumbnail preview of what the website will look like. We have decided to use **BOOTSWATCH**, which will allow more flexible theme selection than **DEFAULT** bootstrap.



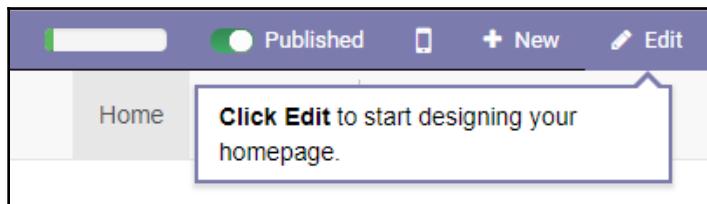
The Website Builder installed for this chapter was **Odoo 12 Community Edition**. Depending on your specific Odoo build, you may have an alternative selection of themes, or you may, in fact, get taken directly to your Odoo home page.

Also, you should be presented with a Website Builder tutorial that will walk you through some of the basics of building your website.

Here, we can see the initial web page presented by Odoo:

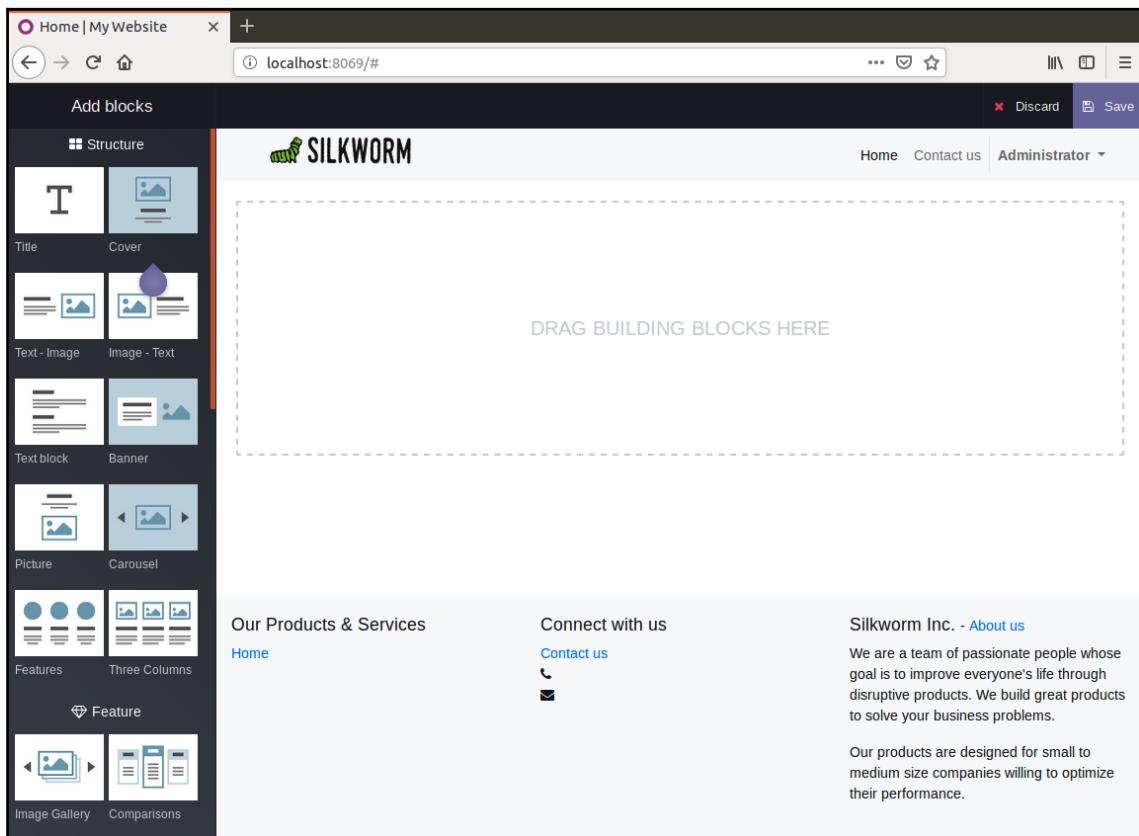


In the middle of the screen, you will see a **Welcome to your Homepage!** title, along with an invitation to follow the blue signs for a simple walkthrough that will assist you in building your first website in Odoo. Note that these tips that walk you through each step are shown just for the tutorial and will only be shown the first time. We will go ahead and use some of these tips to highlight the basic features of the Odoo **Website** application. The following screenshot displays the **Edit** functionality of the **Website** application:



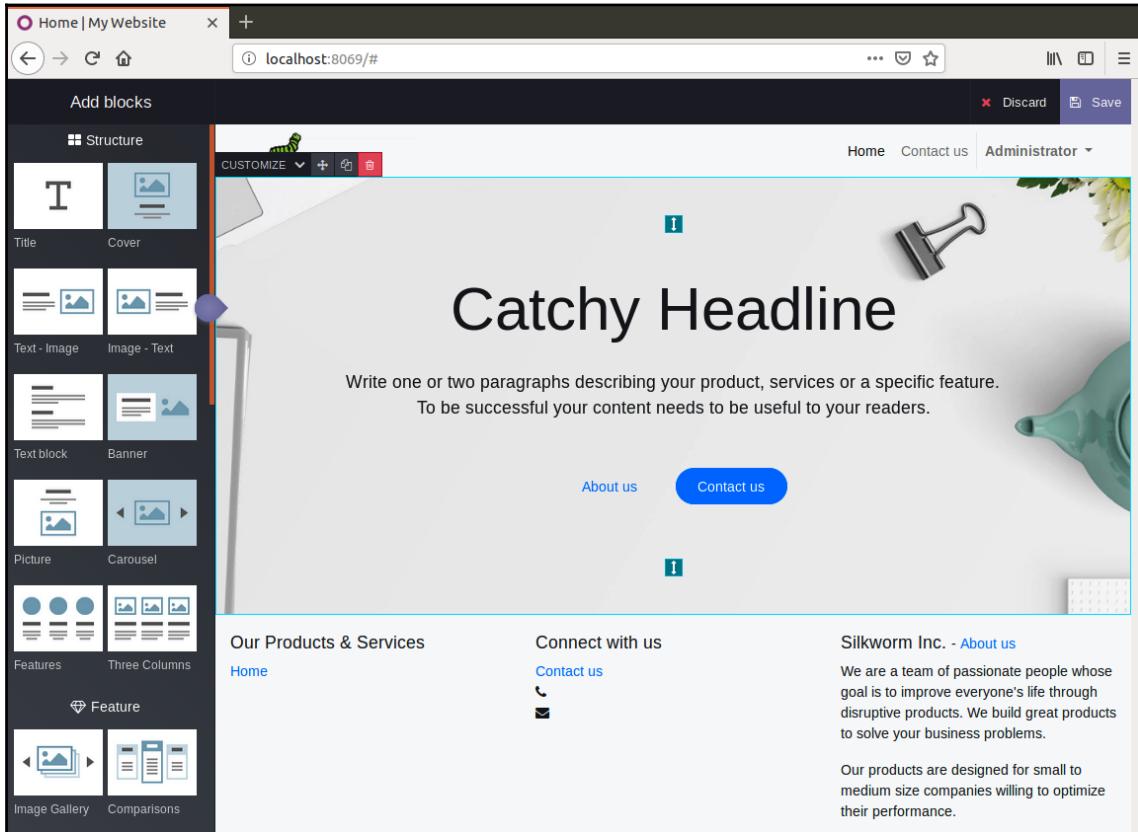
Clicking the  in the top-right corner directs your attention to the **Edit** button at the top-right corner of the screen. This button is available on every page while you are on your Odoo website as an administrator. Clicking this **Edit** button will toggle your page into **Edit** mode so you can make changes to your website. The following are the steps for you to begin editing your home page:

1. Click the **Edit** button to begin editing your home page.
2. After clicking the **Edit** button, your page should refresh to display the toolbars and options available for editing your web page as shown in the following figure:



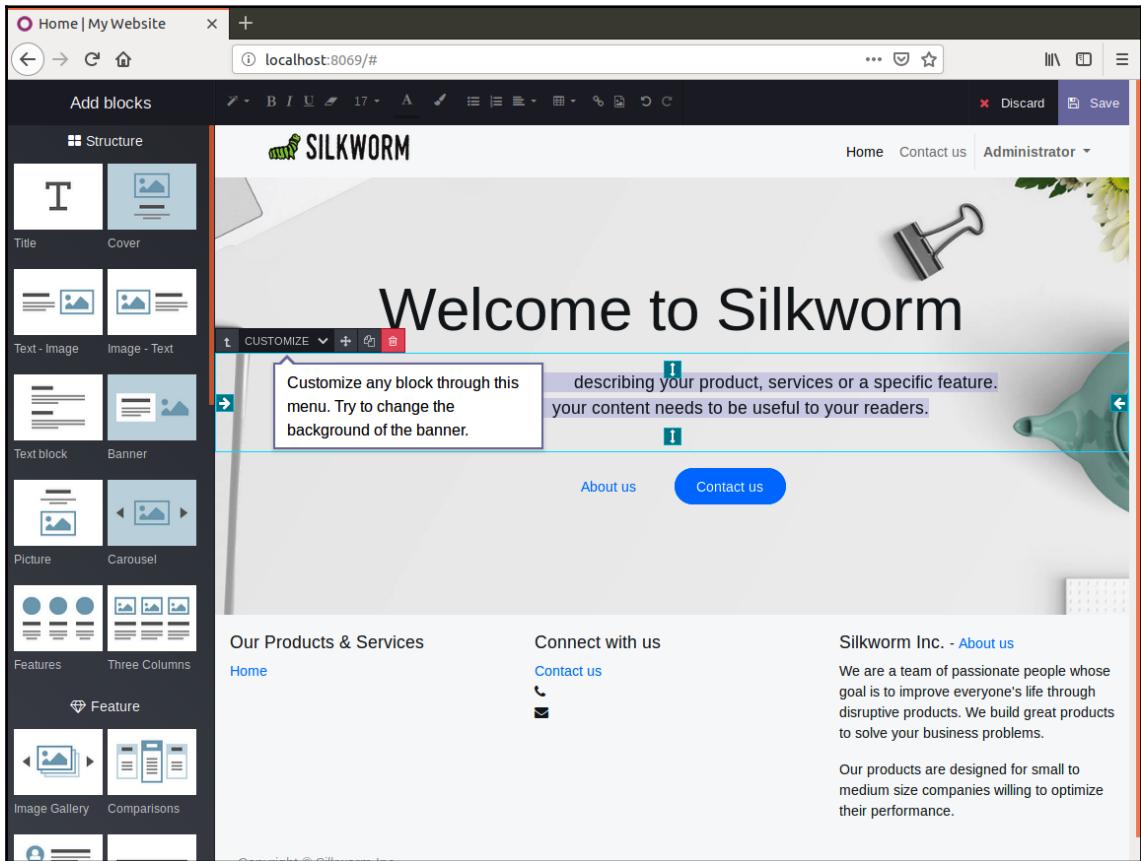
Here, we can see that the tutorial is prompting you to drag the **Cover** block and drop it into your page. This is the action you will take any time you want to add additional content to your web page.

3. Click and hold your left mouse button over the **Cover** block and drag it out onto your web page. The web page will immediately update to show you the cover:

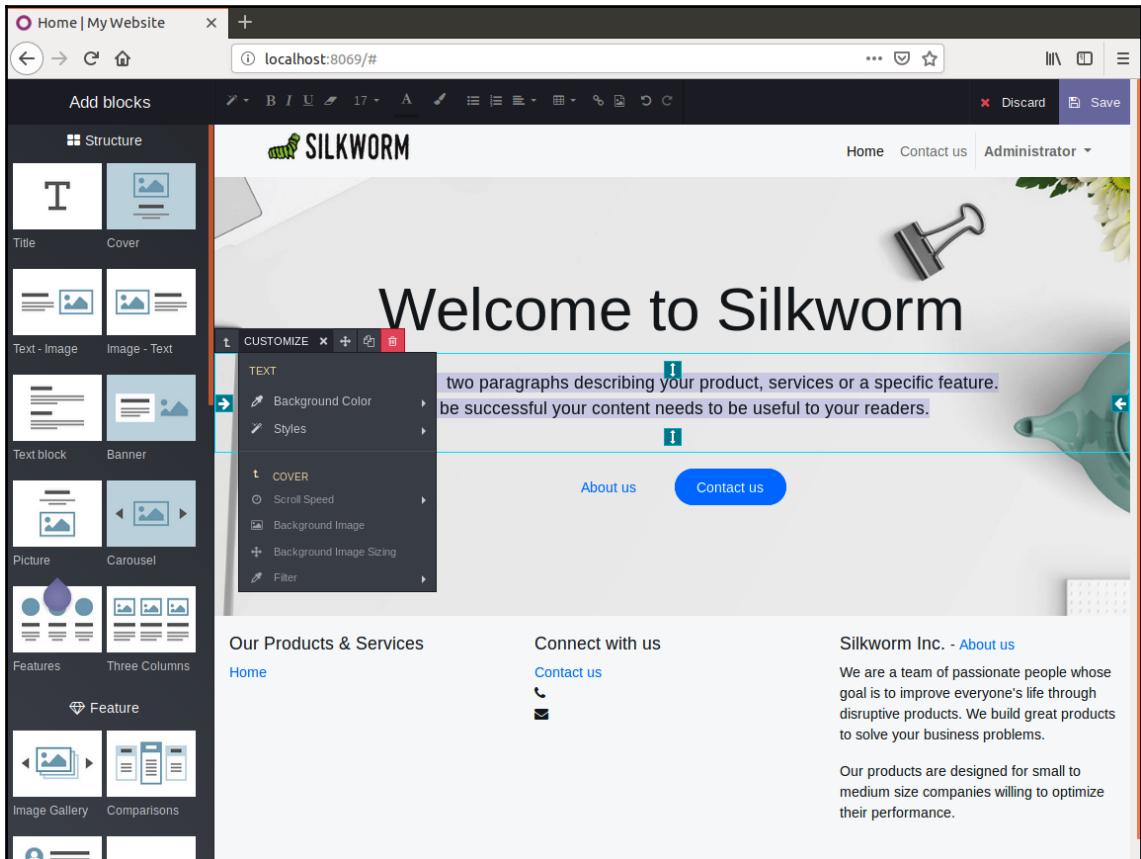


4. Odoo's simple tips will continue as you follow along, prompting you now to change the title of your text to whatever you choose. All the editing is performed right on the page itself. Just use your mouse to select the text, like you would in any simple text-editing program. Also note that, in the top toolbar, you have the ability to modify the text to make it bold, underlined, or italic, or to change another property, such as color.

- After a few changes, the Odoo Website Builder tutorial will come back once again, calling attention to the **CUSTOMIZE** toolbar menu at the top-left corner of the block. This is a very important menu that allows you to edit the properties of the container, as well as to select the parent container of any object you are editing:



6. Following the tip, we can now use the **CUSTOMIZE** menu to change the background of the banner. There are several options, including uploading a graphic or changing the background color:

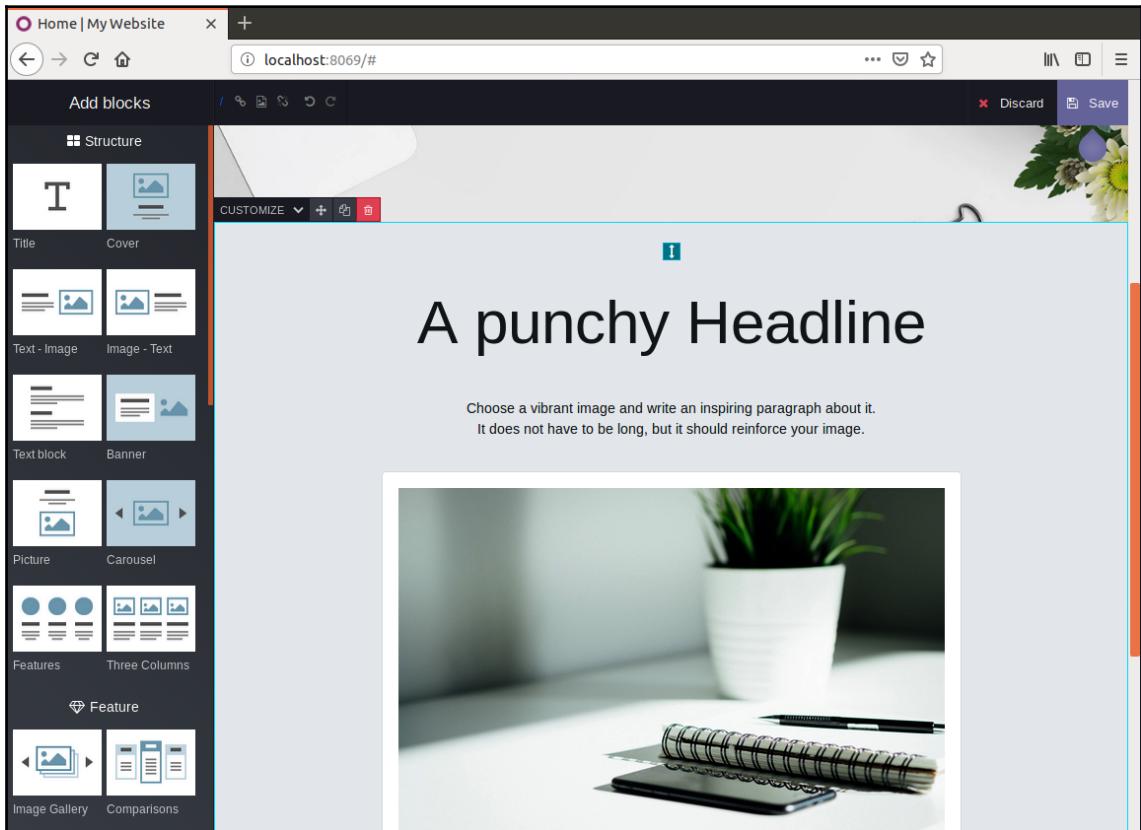


7. As you mouse over the colors, you will see the background of the textbox change to reflect your choice. Pick a background that you like, then the Odoo tip will prompt you to drag an **Image-Text** block onto the page:

The screenshot shows the Odoo website builder interface. On the left, a sidebar titled "Add blocks" displays various block types: Title, Cover, Text - Image, Image - Text, Text block, Banner, Picture, and Carousel. A tooltip from the "Image - Text" block is visible, stating "Drag another block in your page, below the cover." The main content area shows the "SILKWORM" homepage with a large banner featuring a green stamp and a white flower. Below the banner, the text "Welcome to Silkworm" is displayed. A callout box with the same tooltip is overlaid on the sidebar. At the bottom of the sidebar, there are links for "Features", "Three Columns", "Home", "Image Gallery", and "Comparisons". The footer contains sections for "Services", "Connect with us" (with icons for phone and email), and "Silkworm Inc. - About us". It also includes a statement about the company's mission and products.

8. Follow the instructions and drag the block below the Cover block.

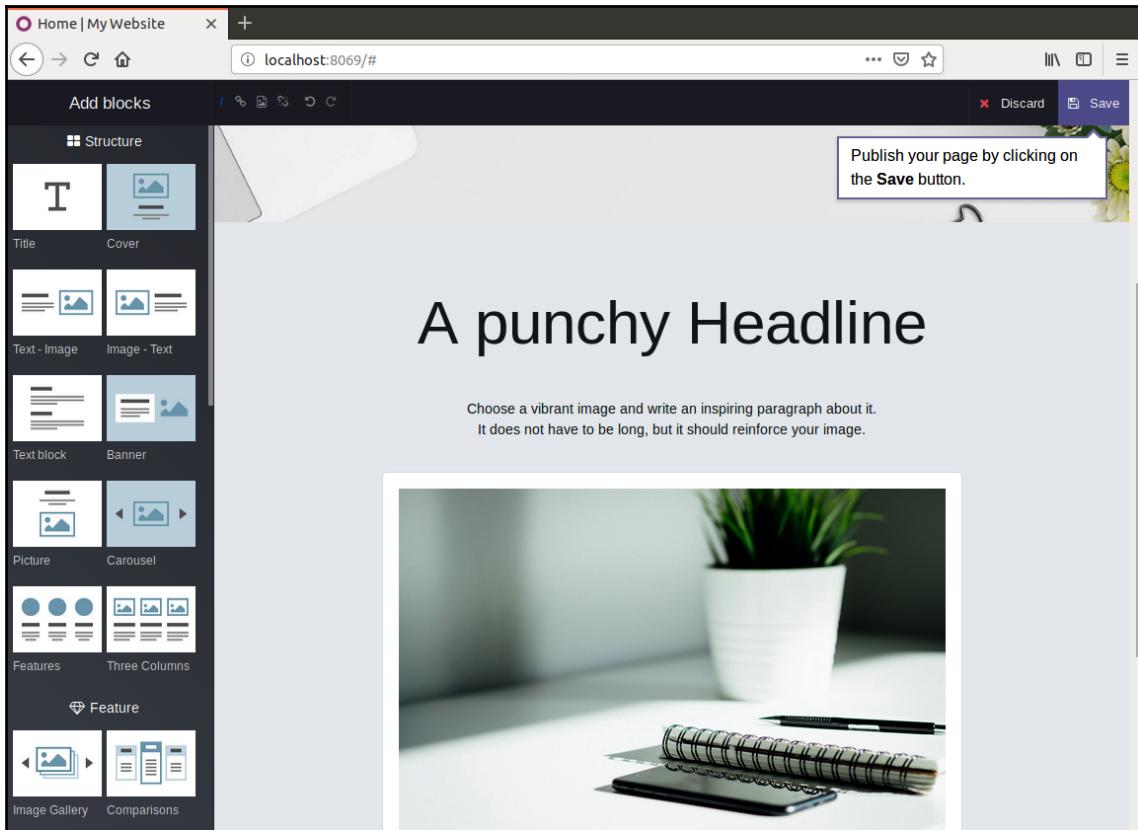
Feel free to change the block as you wish and practice with the various options that are available. The following screenshot is one such example with the use of various options available:



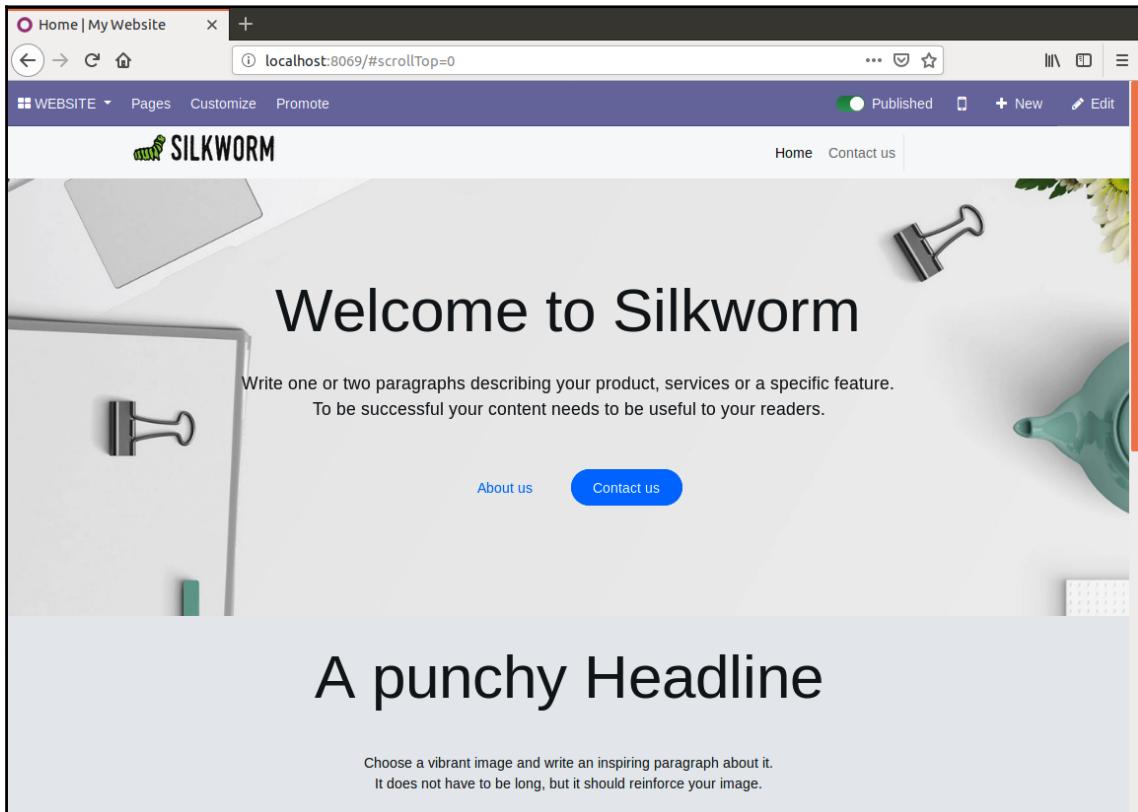
With this, we have successfully installed the application and will be looking into the further enhancement of pages.

Saving your web page

Once you have had a chance to practice editing your web page, use the **Save** button in the top-right corner of your page to publish your changes:



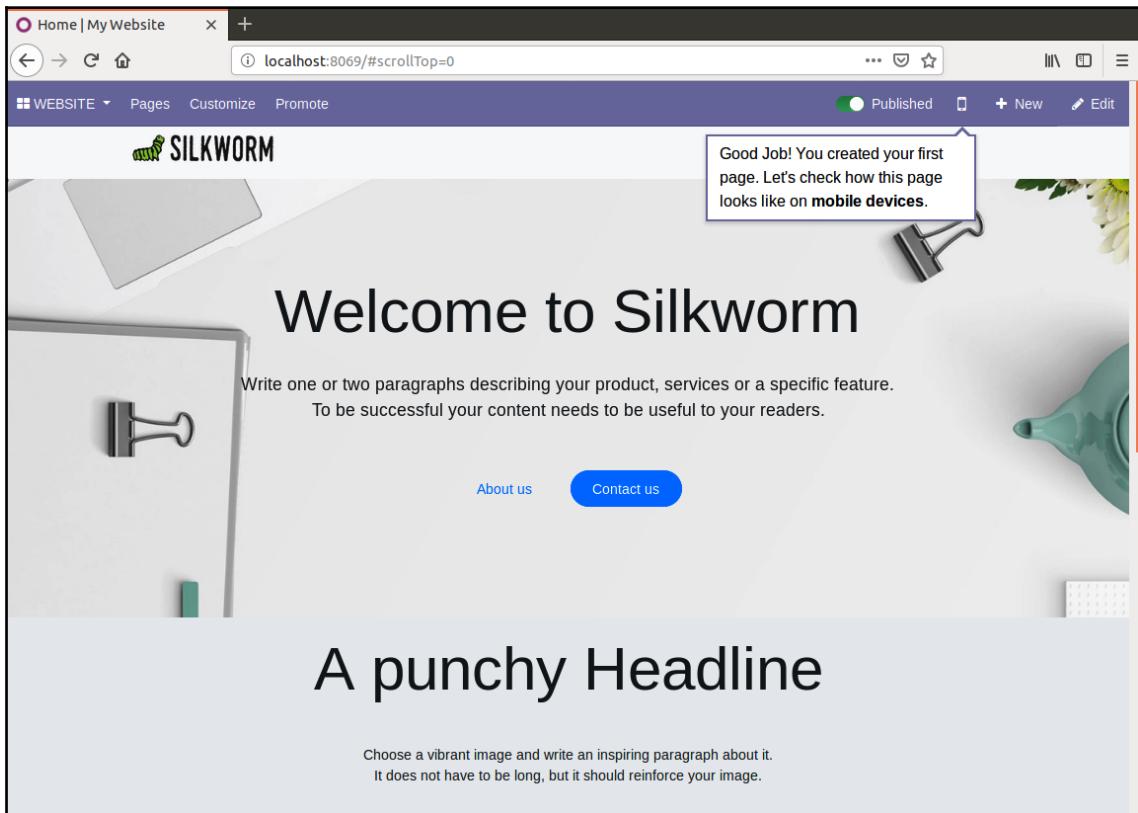
After you click the **Save** button, the page will refresh, and we can see our web page as it will appear to the visitors that come to our site, except for the blue bar at the top, which is only visible when you are logged in to Odoo as a user:



Odoo's tutorial will give you a little congratulation message for successfully saving your website and completing this part of the tutorial. We've now learned how to edit our web pages, add new content blocks to those pages, and save those changes. As you can see, it is pretty easy to create web pages in Odoo using the Website Builder. Still, expect to spend a bit of time learning how the various objects can be combined and edited in order to get the results you desire.

Previewing our website on a mobile device

Certainly, one of the compelling reasons to consider using Odoo's Website Builder for your CMS is that it was built from the beginning to support mobile devices. This feature is so central to the **Website** application that there is a dedicated phone preview button that will let you preview what the website should look like on a mobile device:



Click the **Mobile Preview** button to see how your web page will look on a typical mobile device:

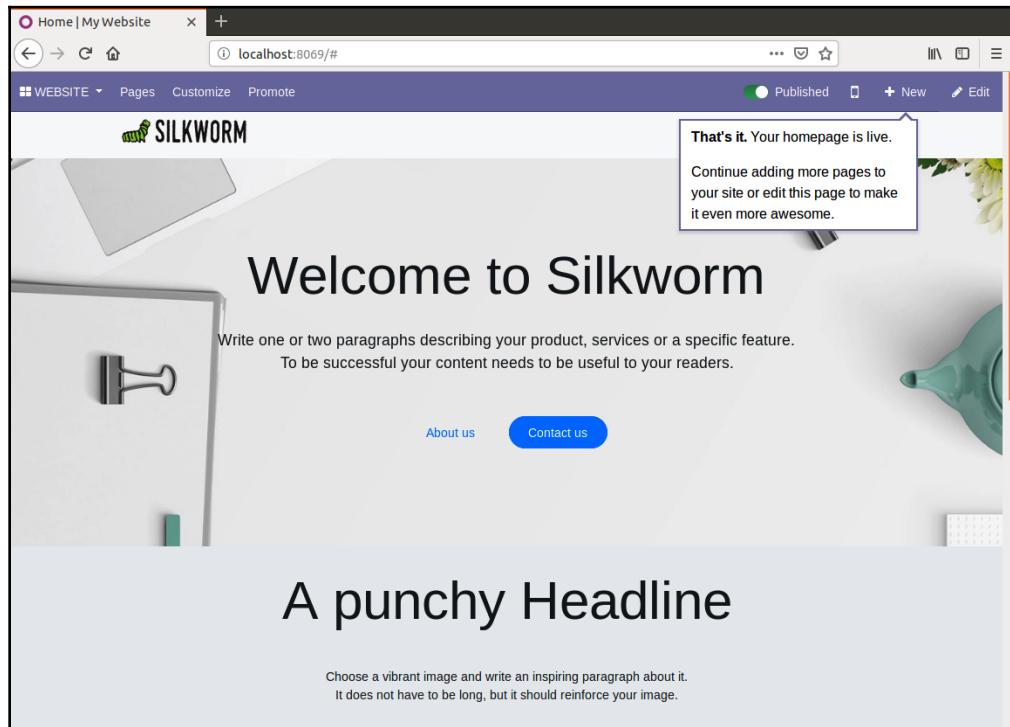


While you should still double-check all of your web pages on real mobile devices before you deploy your website, this feature is very valuable because it allows you to get an idea of what your pages will look like on mobile devices.

Adding new pages and menus to your website

Next, it is time to take a look at how we add new pages and menus into our website. Here, it is a good idea to lay out your ideas ahead of time and decide what pages you need and how the menu structure for your website should look. I personally believe that it is better to start out simple and add additional complexity as you go. Still, there are some pages you will already know that you must have on your website, so you may as well go ahead and add them.

From Odoo's top menu, use the **New** button to add menus and pages, or the **Edit** button to make changes to existing pages:



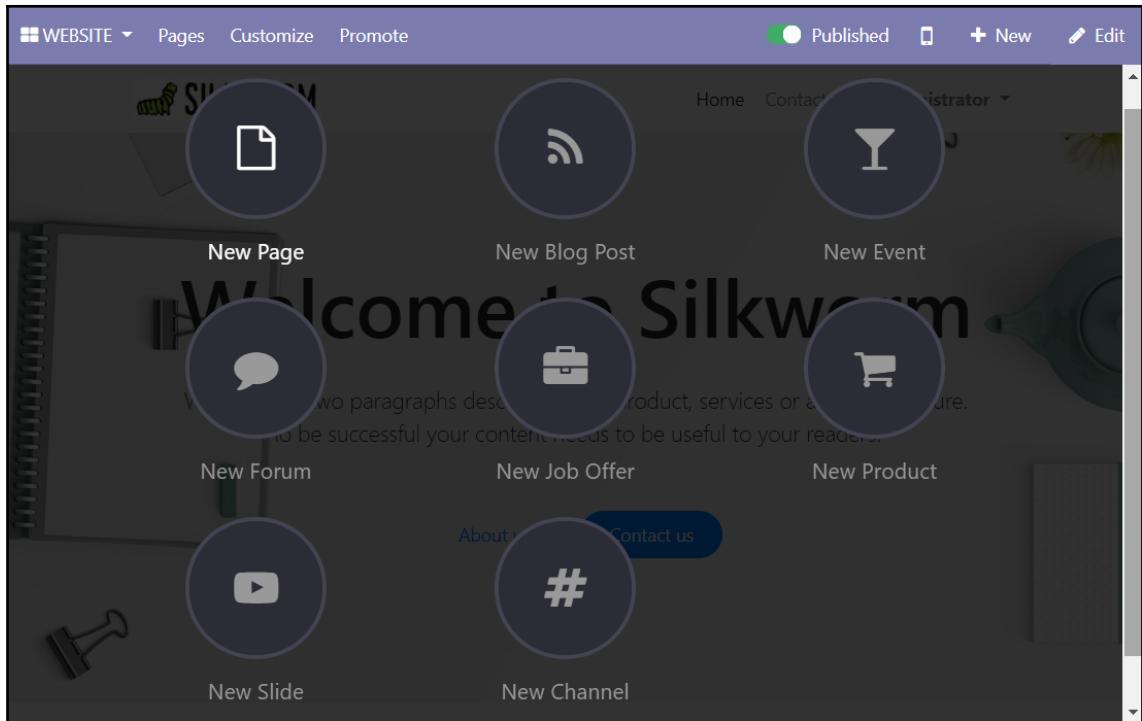
Some pages that are wise to add right away are your **terms of use** and **privacy policy** pages. Every website should have these, and if you've had a website in the past, you likely already have text available to you that you can quickly copy and paste into place.



Odoo **Website** Builder automatically creates the **About Us** and **Contact Us** sections of the website for you, but you will certainly want to edit them both to change the default graphics and to make the text relevant to your company. The **Contact Us** page will even prompt you to install a contact form app to make the page more useful to your website's visitors.

Adding a new page

Clicking **New** brings up a large icon in the center of the page to confirm that you wish to add a new page to your website:



In this example, we will name our page Frequently Asked Questions, a page that is common on many websites. We will also leave the **Add to menu** checkbox marked so that our page will be added to our menu automatically when the page is created:

New Page

Page Title: Frequently Asked Questions

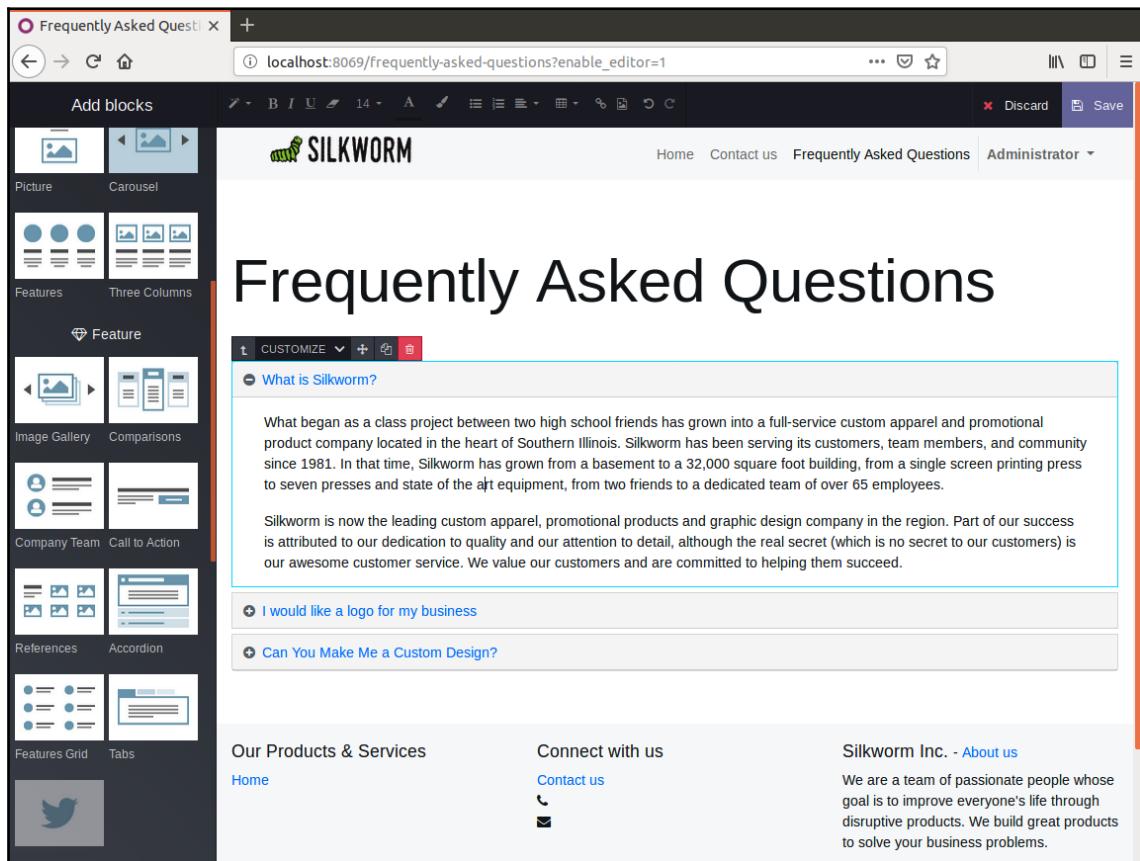
Add to menu

Continue [Cancel](#)

Simply click the **Continue** button and the web page will be added and will be ready for editing, the same as when we modified the previous web page.

Creating our frequently asked questions web page

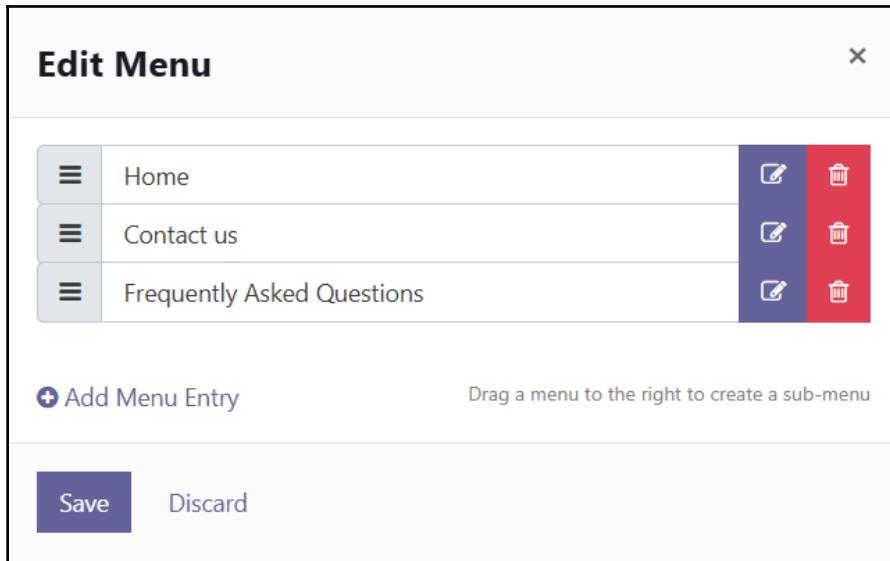
In the following example, after creating the page, we inserted an **Accordion** block, which is found under the **Feature** section in the **Blocks** area. We then edited the block with some of Silkworm's frequently asked questions content as shown in the following screenshot:



Once the page is the way you want it, click the **Save** button to publish your changes.

Managing menus on your website

In the **Pages** menu, you can use **Edit Menu** to organize the menu structure for your website:



Using this simple **Edit Menu** form, you can reorganize your menus by using the sliders on the left-hand to click and drag them to the order you desire. Any menu that you drag to the top will automatically become the home page of the website.

Dragging a menu to the right will nest menus within other menus. Using these basic methods, you can create a hierarchy of menus to contain your pages in any structure you choose.

Adding a new menu

Click the **Add Menu Entry** link to bring up a form that will allow you to add a new menu item to your website.

In this example, we have added a new menu named Design Tool and specified an external link to connect directly to the design tool that Silkworm currently uses, for people who wish to design orders on their website:

The screenshot shows a modal dialog titled "Create Menu". It has two input fields: "Menu Label" containing "Design Tool" and "URL or Email" containing "https://www.silkwormink.com/design-tool/". A hint below the URL field says "Hint: Simply type '#' to create a container menu". At the bottom are "Save" and "Discard" buttons.

Menu Label	Design Tool
URL or Email	https://www.silkwormink.com/design-tool/

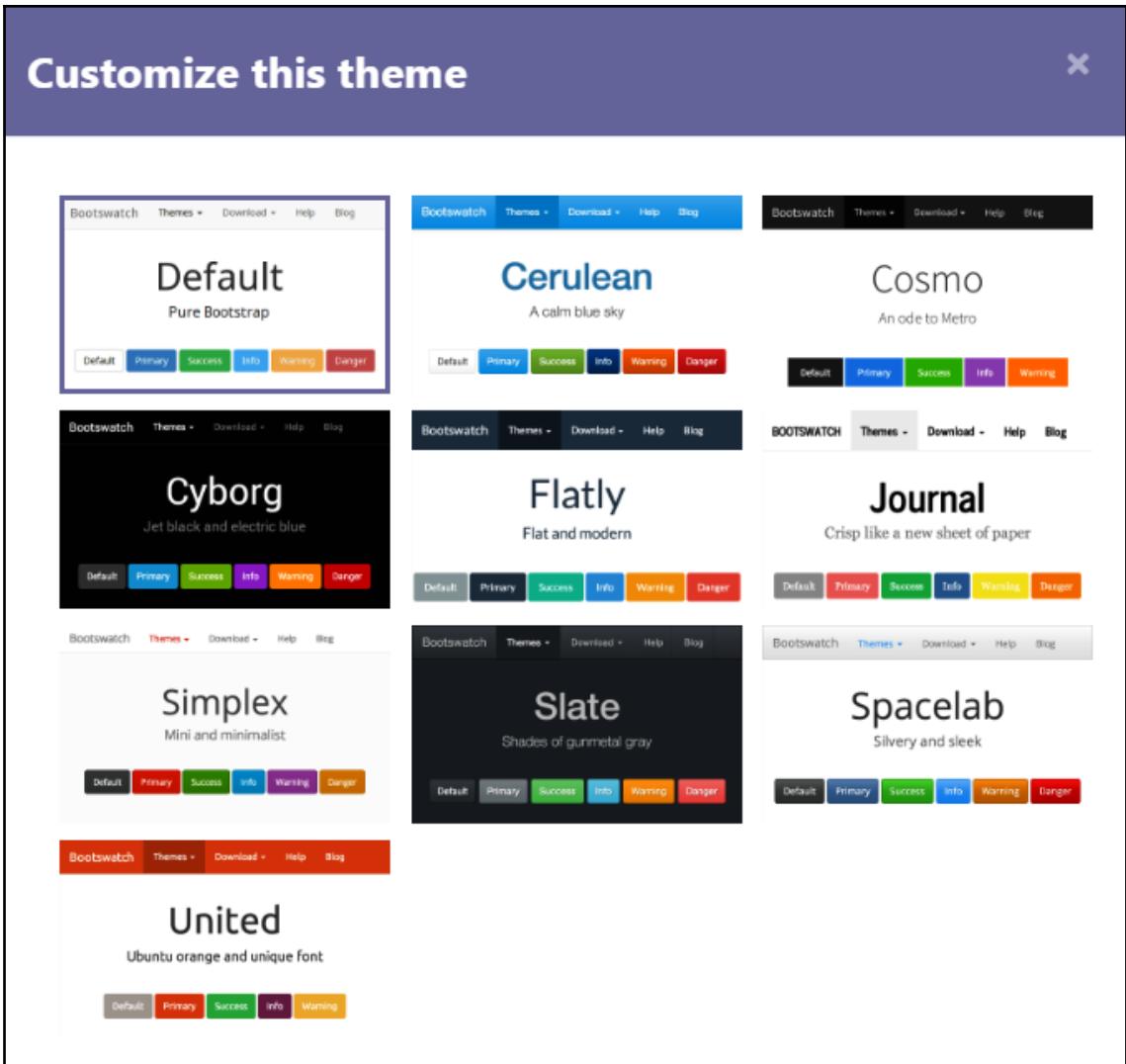
Hint: Simply type '#' to create a container menu

Save Discard

The form allows you to have flexibility in assigning a menu to an existing page, creating a new page, or specifying a **URL or Email** address to link to the menu.

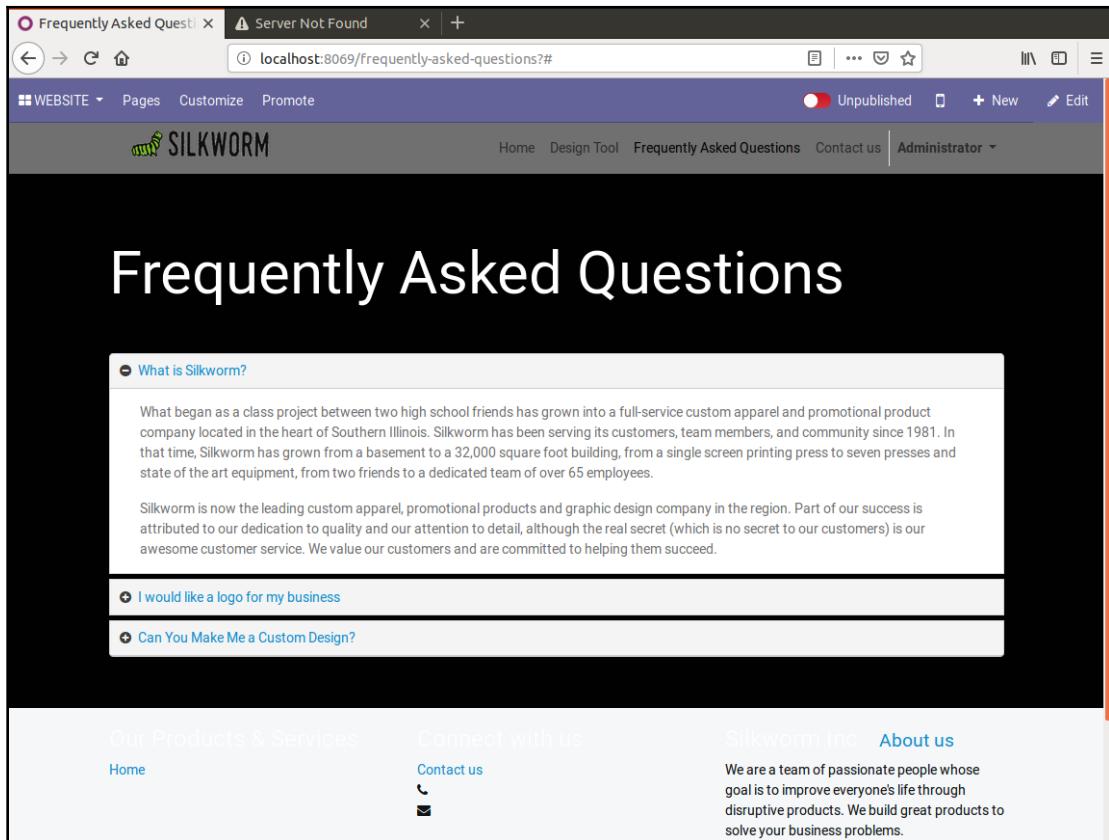
Changing themes in Odoo

One of the attractive features of most CMS solutions is their ability to change the theme of your website without having to modify your content. Odoo's Website Builder provides the ability to modify your theme by selecting the **Customize Theme** option from the **Customize** menu:



With this build of Odoo 12, you have the option to select from a variety of free bootstrap themes. Simply click **Apply**, and your website will then be updated with the new theme.

Here, we can see how our **Frequently Asked Questions** page looks after we have applied the **Cyborg** theme and adjusted the font color of the main body text:



As you can see, simply by changing the theme of your website, you can create a dramatically different look.

Promoting your website

Some of the other nice features of Odoo's Website Builder are the built-in promotion tools for your website. It is not only helpful to use the promotion option for each of your web pages, it is vital. If you don't go in and specify the title, keywords, and descriptions for your pages, Odoo will provide default information to search engines such as Google. This is never a good idea! Be sure to take the time to at least provide a proper title and description for each of your web pages.



Search engine optimization (SEO) refers to practices used to build good website traffic by making your website easier to find within the most popular web search engines. No successful business wants their website to linger in the rear of the search results list while competing websites are jumping to the front of the line and garnering all of the customer views.

To promote your web page, go to the web page you wish to promote, click **Promote** in the **Website Builder** menu, and choose **Optimize SEO**. Odoo will then bring up the promoting form as shown in the following screenshot:

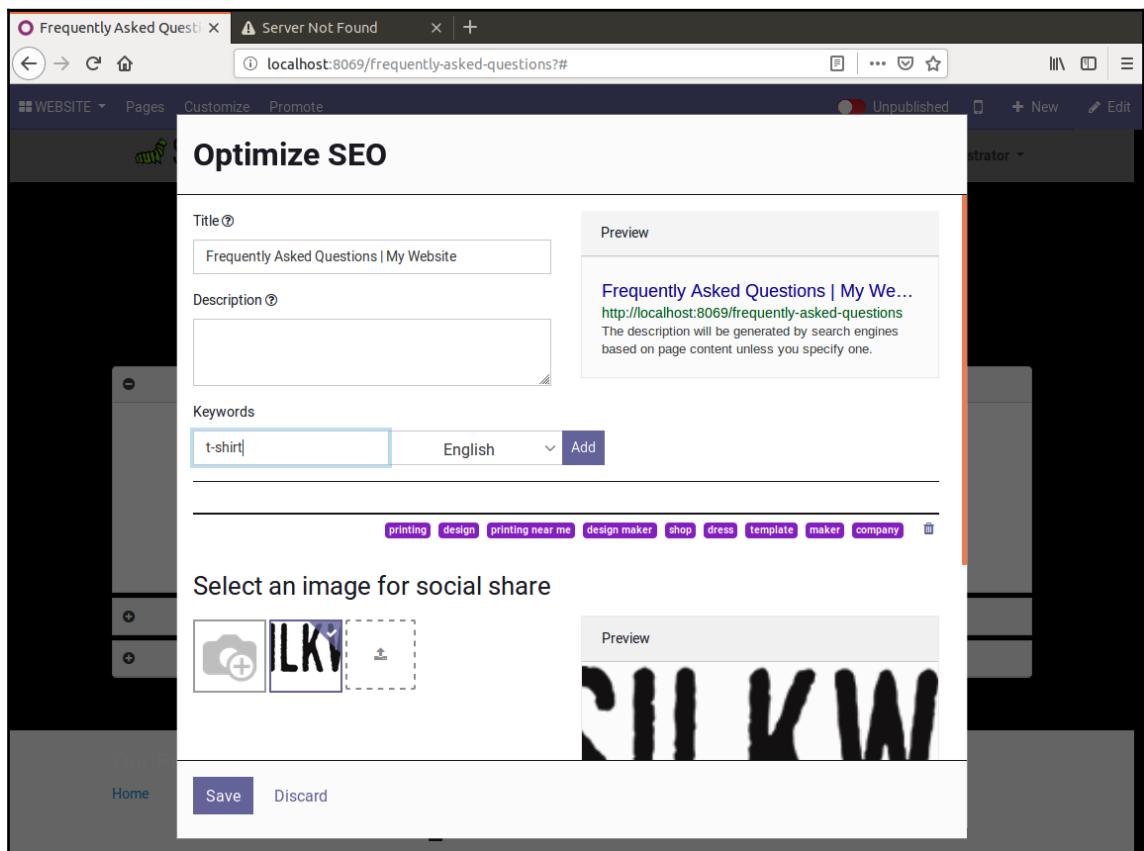
The screenshot shows the Odoo Website Builder interface with the 'WEBSITE' tab selected. A modal window titled 'Optimize SEO' is open for the page 'Frequently Asked Questions | My Website'. The form includes fields for Title (set to 'Frequently Asked Questions | My Website'), Description (empty), and Keywords (empty). A preview section shows the page title and URL. Below the form, there's a section for selecting an image for social share, featuring a placeholder image of a camera icon and the letters 'ILK'. At the bottom, there are 'Save' and 'Discard' buttons.

For this example, we have intentionally left the fields as they were when the form was brought up so that you can see why it is so critical that you specify the title and description for each of your pages. Note how Odoo has named our page, and that the preview at the bottom is nothing like what we would want to have listed within search engines.

Specifying keywords for your website

SEO is a huge topic that could fill an entire book all on its own. One of the major aspects of good SEO is that it knows what keywords are most popular for the page you wish to promote. Odoo provides a handy little tool that ensures that as you specify keywords, Odoo will let you know the associated keywords that are also popular with Google.

In the following screenshot, we have entered the t-shirt keyword and can see associated keywords that may be good to include within the content of our web page, in order to get better results in search engines:

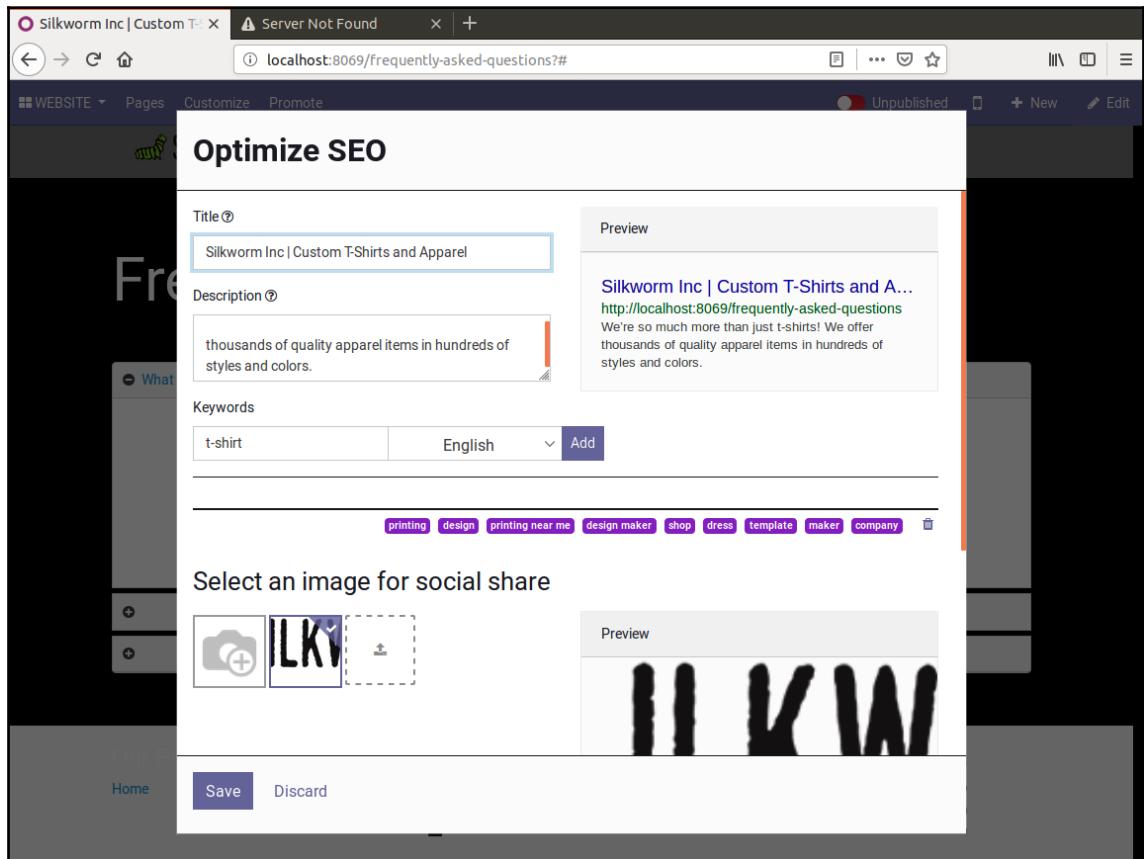


In the results, next to the t-shirt keyword, you will see **printing, design, design maker, dress, template, maker**, and more. Odoo also indicates which keywords, if any, are already located within our web page.

Creating a title and description for your website

Take time to make a description that includes good keywords and accurately describes the content of your web page. There are many good books available to help you better promote your website and provide advice on how you can get the best results in search engines.

In the example shown in the following screenshot, I have added a few keywords to the title as an example of how Odoo will color the keywords you have selected. We have also turned the theme back to the default to make it easier to read the text. The purpose is to try to have as many keywords as possible in your title and description so that you can get better results for your page. In this case, **design** and **printing** are important keywords that I have included in the title based on the feedback from Odoo's keyword research:





Note that this is only an example; it is far more important to have accurate descriptions than to simply make a title based on popular keywords.

Summary

In this chapter, we took a look at Odoo's exciting new Website Builder. We discussed CMSes a little bit, as well as some of the other more popular options. We then demonstrated how to access the **Website** application and followed along with Odoo's simple but effective interactive tutorial to learn the basics.

Then, we learned how we could add new pages to our website, configure and edit menus for the website, and add additional blocks and content to our pages. Finally, we finished up by learning how to change themes to give our website a new look, as well as how to promote our web pages for good search engine results and for a proper preview and description within Google.

In the next chapter, we will look at how we can use our new website to host an e-commerce shopping cart that integrates directly with our products in Odoo.

12

Implementing E-Commerce with Odoo

In the previous chapter, we looked at the new **Website Builder** application and saw how it can be used to create a website with ease. Now that we understand the basics of getting a website up and running, this chapter will show you how you can extend a simple website to become a full eCommerce site that can take and manage orders. Even better, this functionality ties directly into sales management, which you learned about in Chapter 3, *Exploring Customer Relationship Management in Odoo 12*.

In this chapter, we will cover the following topics:

- An overview of commerce and how it is implemented in Odoo
- How to install eCommerce and view the default web store
- Modifying your online store to contain your products
- How to use product variants such as color and size to provide customized options for the products you sell
- Advanced product options such as alternative products, accessories, and categorizing your product
- How to set up a payment processor

eCommerce and Odoo

As you will probably be aware, eCommerce is a term that describes offering your products and services to customers electronically, typically on a website. Over the years, eCommerce has expanded to include more and more markets. There are many ways to take payment on websites, and now many mobile applications include the ability to take micro-transactions. eCommerce now covers a very wide field of options.

Popular eCommerce platforms

Despite Odoo's late entry into the Website Builder/**content management system (CMS)** market, Odoo is introducing eCommerce functionality into a mature market with several options. While there are hundreds, if not thousands of viable eCommerce platforms out there. The following are a few of the more popular ones.

Magento

A list of popular eCommerce applications would not be complete without Magento. Magento is open source and very popular, and is often a user's first choice when integrating with Odoo. Even with Odoo's new eCommerce option, Magento continues to be popular with those who need more advanced functionality.

Volusion

Volusion is also a very popular eCommerce platform that takes an entirely different approach. Instead of an open source solution, Volusion is a hosted solution in which you configure your website and cart. Volusion offers many different pricing models, as well as a free 14-day trial. While not for all companies, Volusion can be a fast solution for companies that need to get up and running without having to worry about installing software.

Shopify

While Shopify is very popular, and is a hosting solution like Volusion, it also directly markets point of sale solutions as well. This means that Shopify is often a good solution for companies that want to have an online presence but also have retail operations. Like Volusion, Shopify also offers a variety of pricing options and a 14-day free trial.

Yahoo Small Business

Also a hosted solution, Yahoo Small Business offers a straightforward online shopping cart; in other words, what you would expect in a basic eCommerce solution. With Yahoo, advantages include easy advertising integration, local marketing options, and robust email-handling options.

Odoo as an eCommerce platform

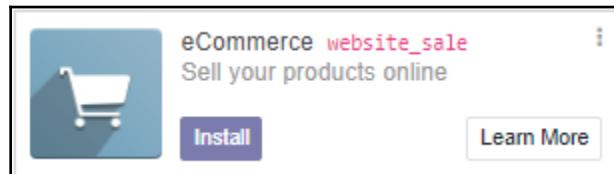
In Odoo, eCommerce is implemented within the Website Builder; the Website Builder is a dependency in Odoo eCommerce. Odoo will install the Website Builder if it is not already installed in the database. In addition to adding a great deal of functionality to the Website Builder, eCommerce also provides additional options that are applicable to products, such as variants, the ability to appear in multiple categories, associations with alternative products, and so on.

While other eCommerce platforms may have more features, there are some significant advantages to using Odoo eCommerce – especially if you are already using, or are planning to use, Odoo as your primary accounting/ERP system. For example, products you enter into Odoo automatically integrate with your eCommerce website, and orders you receive will automatically come into Odoo.

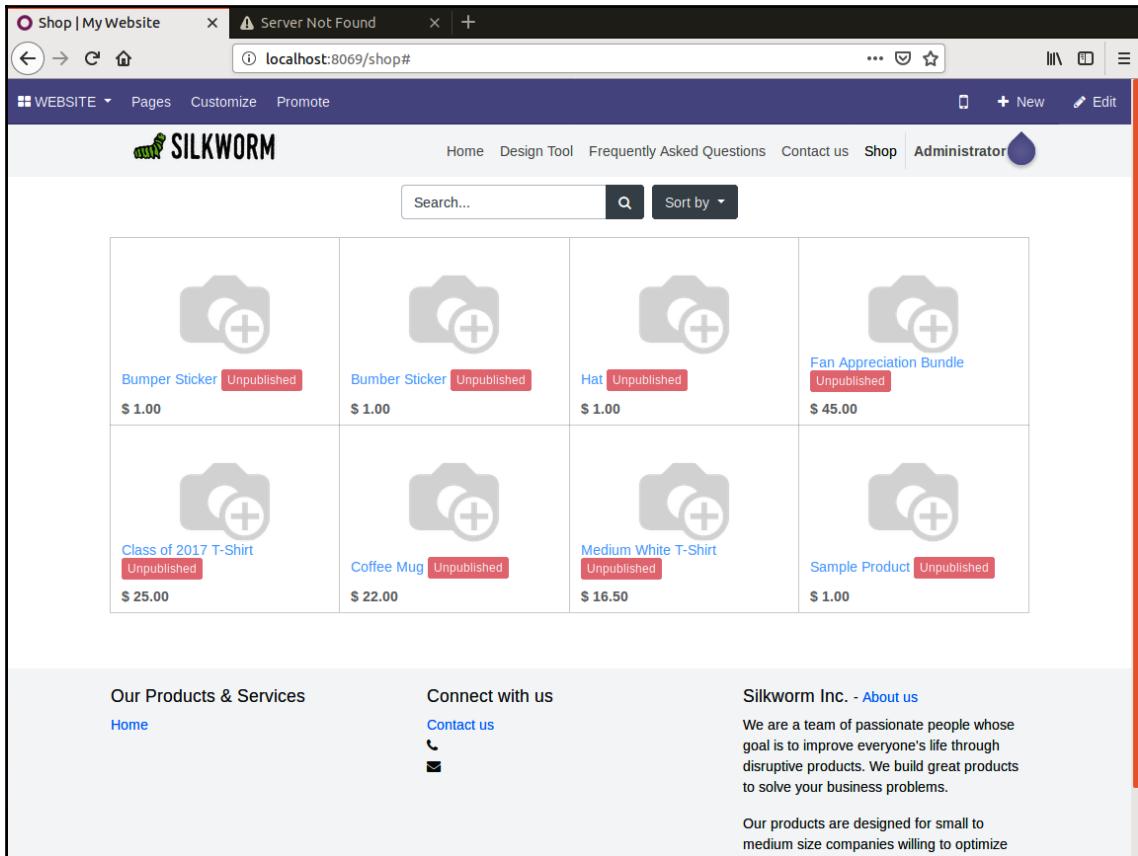
Best of all, Odoo eCommerce is extremely easy to set up.

Installing Odoo eCommerce

You can install Odoo eCommerce like any other Odoo application as shown in the following screenshot:



After clicking **Install**, Odoo will refresh the browser and you will be taken to the store page that Odoo has added to the **WEBSITE** menu. Odoo has a tutorial that takes you through the basics of the Odoo website. Here, you should see the basic Odoo shop setup that has been displayed using the data we've entered throughout this book:



The screenshot shows the Odoo Shop interface. At the top, there's a header bar with a search bar containing 'Search...', a magnifying glass icon, and a 'Sort by' dropdown. Below the header is a navigation bar with links for Home, Design Tool, Frequently Asked Questions, Contact us, Shop, and Administrator. The main content area displays a grid of eight products:

 Bumper Sticker Unpublished \$ 1.00	 Bumper Sticker Unpublished \$ 1.00	 Hat Unpublished \$ 1.00	 Fan Appreciation Bundle Unpublished \$ 45.00
 Class of 2017 T-Shirt Unpublished \$ 25.00	 Coffee Mug Unpublished \$ 22.00	 Medium White T-Shirt Unpublished \$ 16.50	 Sample Product Unpublished \$ 1.00

Below the grid, there are three sections: 'Our Products & Services' (with a 'Home' link), 'Connect with us' (with links for 'Contact us' and social media icons), and 'Silkworm Inc. - About us' (with a paragraph about the company's mission and a note about product design).

You will notice that there is a placeholder showing you where the pictures for your product will display once you've added an image. We'll take a look at how to do that a bit later in this chapter.

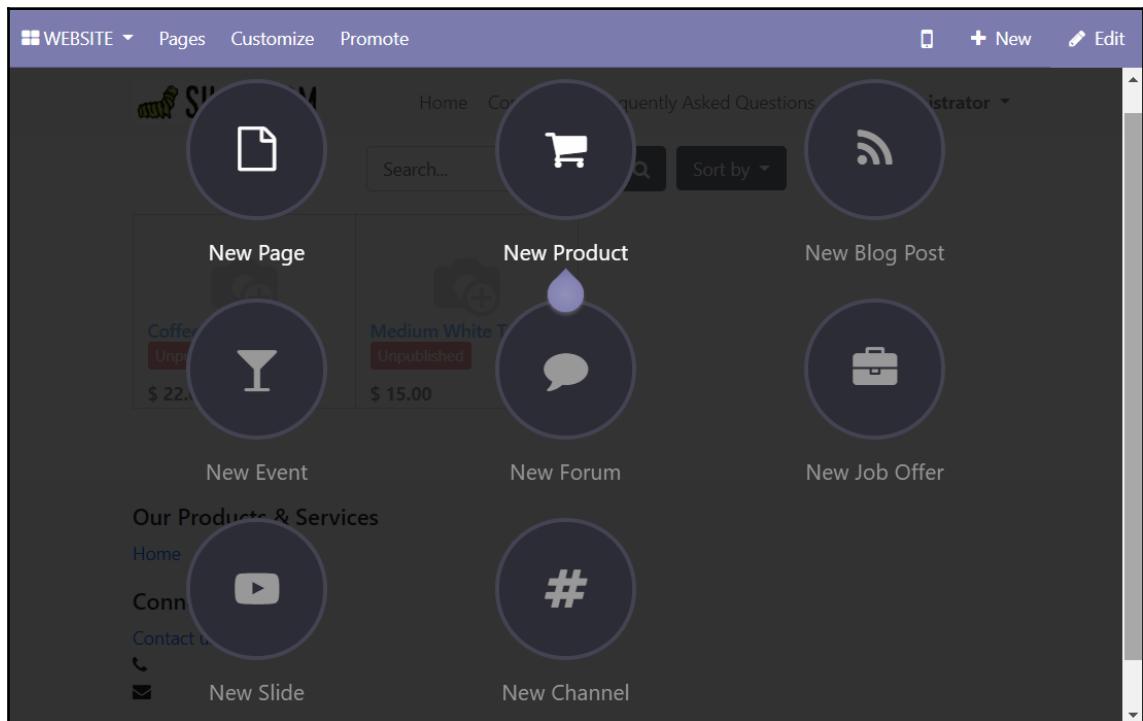
Setting up Odoo is a straightforward task. The easy integration of your products, and eCommerce's relationships with other Odoo applications, means it is a much less complex approach than syncing Odoo to an external eCommerce system.

Basics of Odoo eCommerce

You will notice a few things about Odoo eCommerce right away. There is a search box at the very top of the page that will allow you to search for products on the store. It is also worth mentioning that the footer is shared between all the pages on the website.

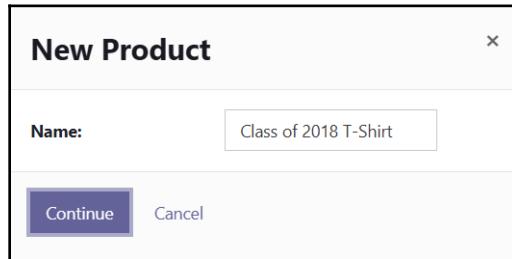
Odoo has a handy feature that allows you to enter products directly from the website. In other words, you don't have to go into the **Sales** menu in Odoo and click and add products there; instead, you can do it right from this shop page.

By clicking the **New** button at the top of the form, you will be prompted to add either a **New Page** or a **New Product**, as shown in the following screenshot:



Let's go ahead and click the **New Product** button to bring up the form.

In this case, we have entered the name of a new product that we want to add to our store, and we have been prompted to click the **Continue** button to add the product:



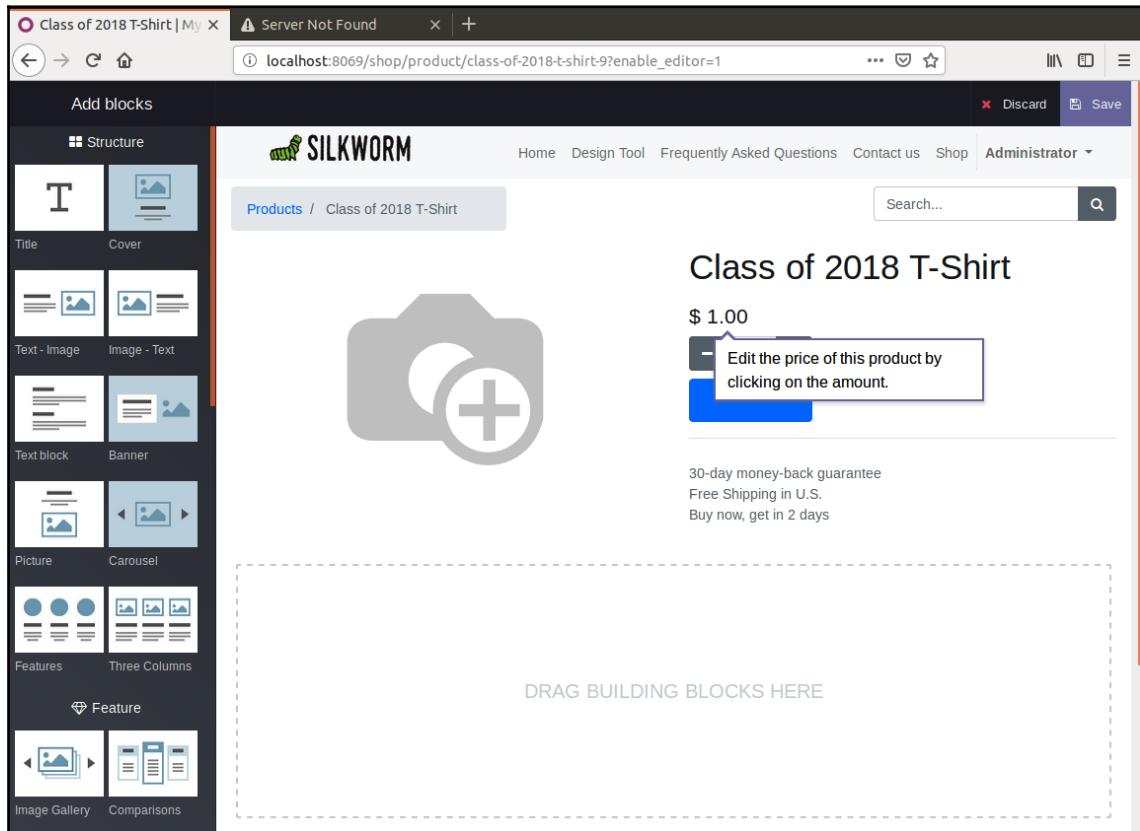
Odoo then refreshes the web browser to show the new product page that your users will see. Here, you can edit and add content to the product page, as you saw in the previous chapter, *Chapter 11, Building a Website with Odoo*, when we talked about the Odoo Website Builder:

The screenshot shows the Odoo Website Builder interface. On the left, a sidebar titled 'Add blocks' lists various content types: Structure (Title, Cover, Text-Image, Image-Text, Text block, Banner, Picture, Carousel, Features, Three Columns, Feature, Image Gallery, Comparisons). The main content area displays a product page for 'Class of 2018 T-Shirt'. The page includes a large image placeholder, the product name, a price of '\$ 1.00', a quantity selector set to 1, an 'Add to Cart' button, and a guarantee notice: '30-day money-back guarantee', 'Free Shipping in U.S.', and 'Buy now, get in 2 days'. A dashed box at the bottom indicates where building blocks can be dragged onto the page. The top navigation bar includes links for Home, Design Tool, Frequently Asked Questions, Contact us, Shop, and Administrator.

While the page is not much to look at yet, you can see that it includes the basics. The price is under the name of the product, as is the option to change quantity, as well as the button that will add that product to a shopping cart. Using what you have learned in the previous chapter, you can drag and drop blocks from the left-hand side to add content to the product page.

Setting the product price from the eCommerce page

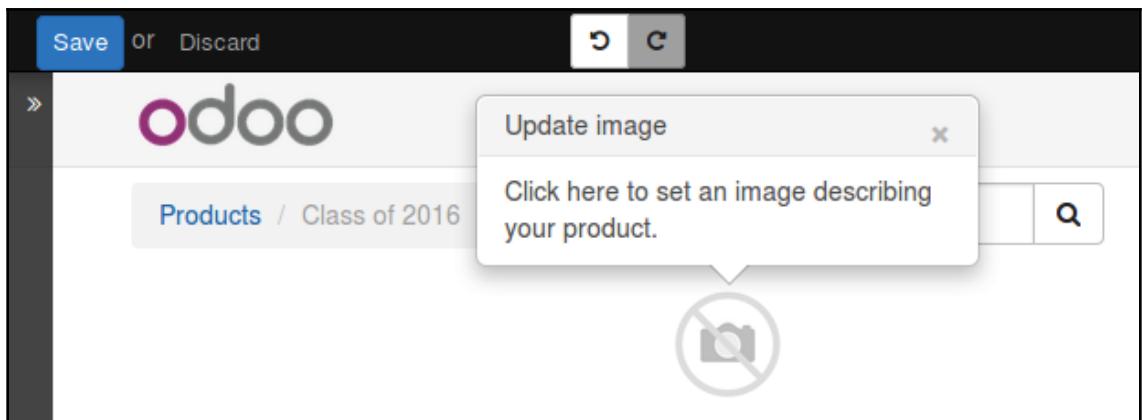
Another nice feature of Odoo eCommerce is that you don't have to go all the way back to the product lists inside standard Odoo business applications. You can change the price on the website, as shown in the following screenshot:



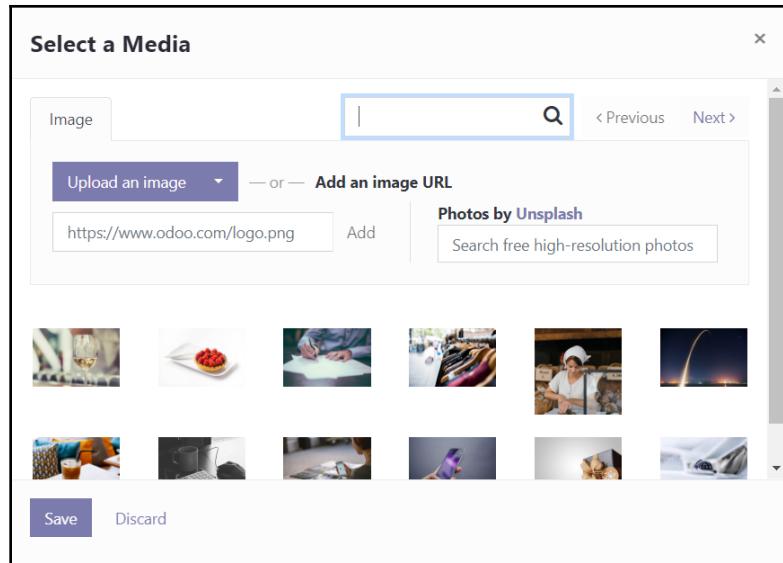
While it may seem like a small thing, the ability to change the price on the web page can help with your workflow.

Adding a picture to your product

If you are going to run an eCommerce site, you will need to have a picture of your product. After following Odoo's tutorial, you will be prompted to add a picture for the product, as shown in the following screenshot:

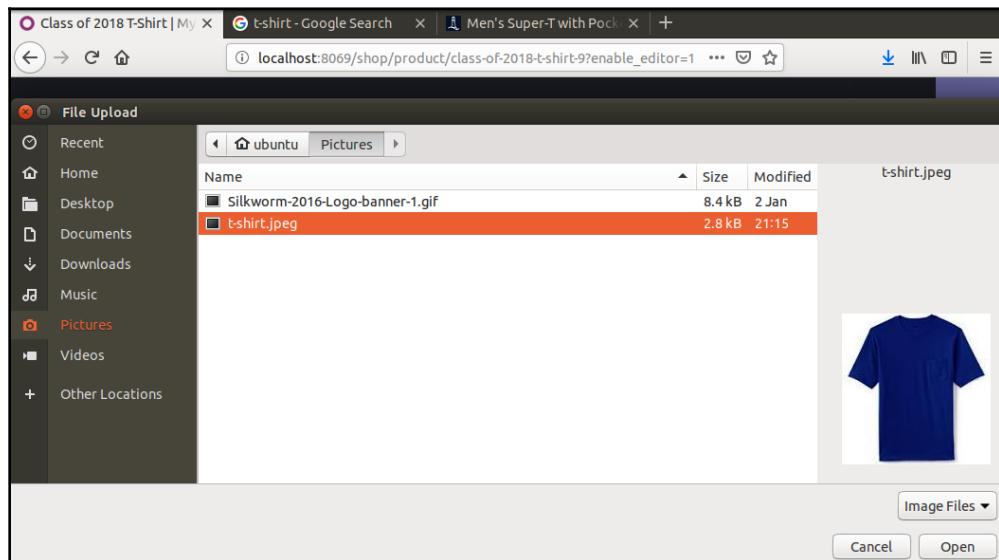


When you're presented with Odoo's picture selection wizard, you have the option to choose pictures from a gallery, your own computer, or even from a URL:

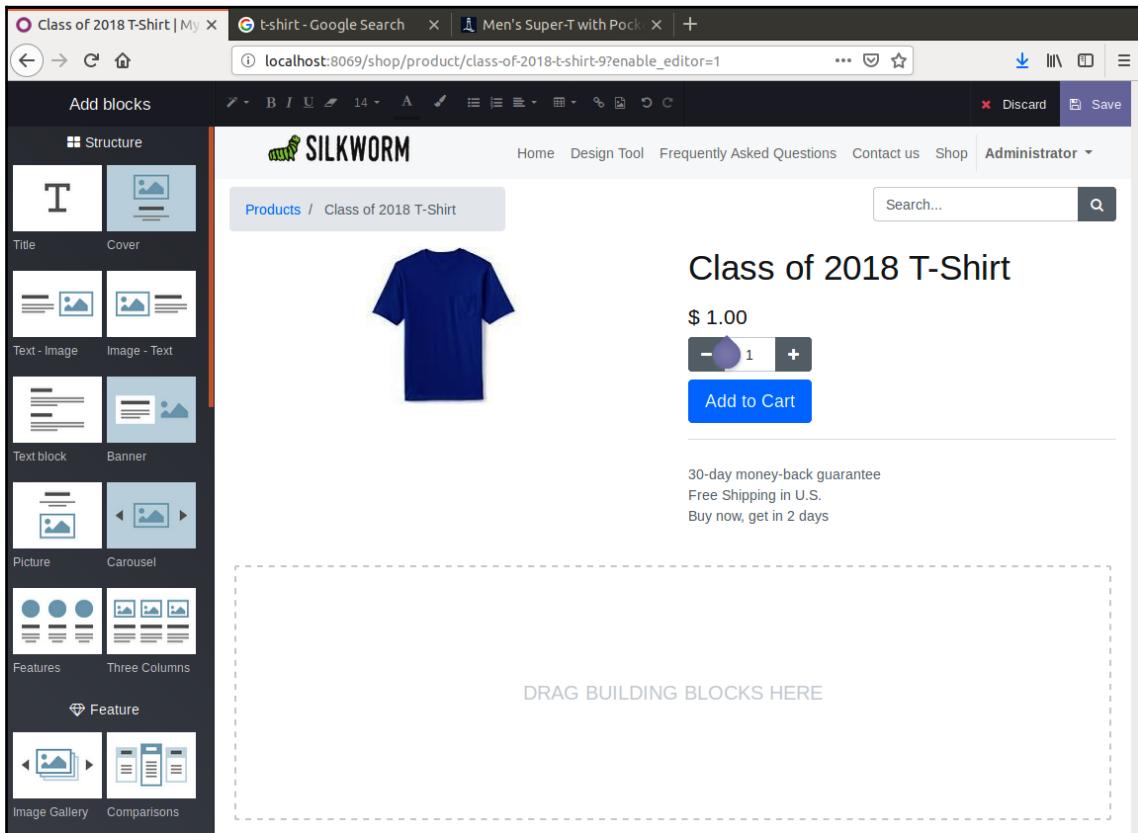


Note the **Upload an image** and **Add an image URL** buttons near the top. You will probably use these more frequently than the built-in gallery pictures that Odoo provides.

We are going to use the **Upload an image from your computer** option to select an image of a simple blue T-shirt. We are using Chrome on Windows 10 to upload `Blue_Tshirt.jpg`, which was obtained from the Creative Commons image library:



Once you click **Open**, the page will refresh and you should see that your product image has been successfully uploaded, as shown in the following screenshot:



Odoo will automatically resize your graphic so that it displays appropriately. If you upload a large graphic, it should reduce the size of it so that it's reasonable for a website.

Describing the product

You should also be able to **Add Blocks** and drag a **Text Block** onto the page. If you need to, take the time to learn what blocks are available and how to use them to make a product page that meets the needs of your customers. You can review Chapter 11, *Building a Website with Odoo*, to learn more about inserting blocks into web pages.

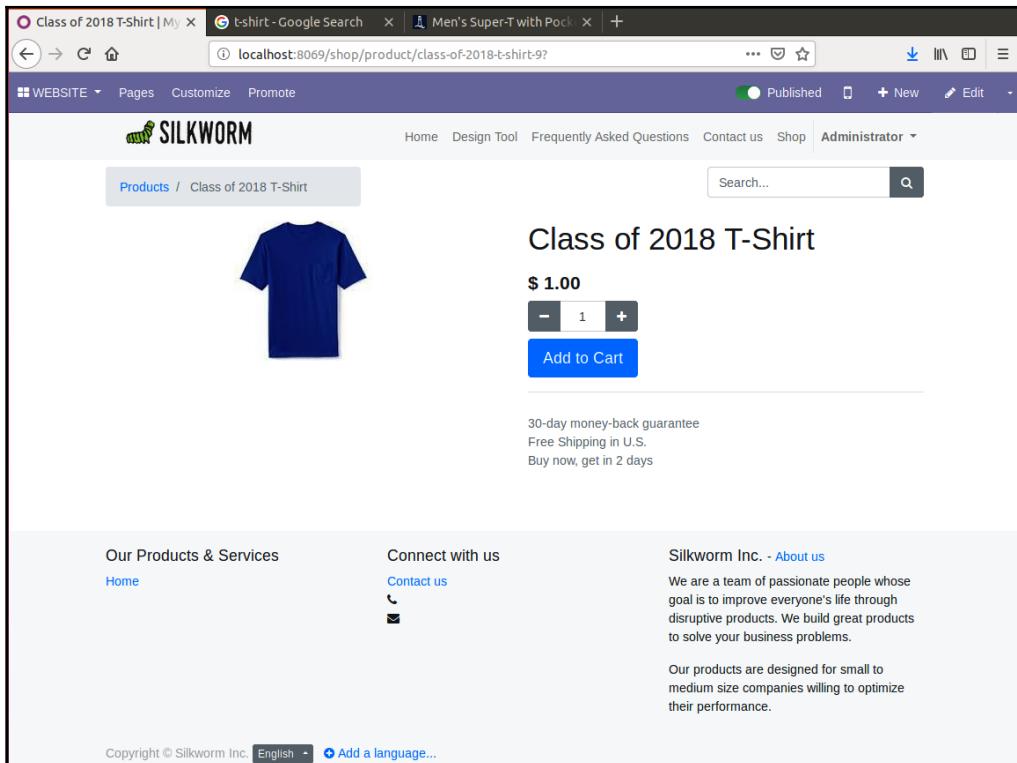
After you have finished inserting blocks and editing pages, click the **Save** button to commit your changes.

Publishing your product

By default, both new and existing products in Odoo are not published to the website. You can see them when logged in either as an administrator or a user with appropriate permission. Until you publish the product, they will not be seen by anonymous website users:

The screenshot shows a web browser window with three tabs open: 'Class of 2018 T-Shirt | My...' (active), 't-shirt - Google Search', and 'Men's Super-T with Pock...'. The address bar shows 'localhost:8069/shop/product/class-of-2018-t-shirt-9?'. The page title is 'SILKWORM'. The main content area displays a blue t-shirt, its price '\$ 1.00', a quantity selector (set to 1), and a 'Add to Cart' button. Below the product details, there is a promotional message: '30-day money-back guarantee', 'Free Shipping in U.S.', and 'Buy now, get in 2 days'. At the bottom of the page, there are sections for 'Our Products & Services' (with links to Home), 'Connect with us' (with icons for Contact us, Phone, and Email), and 'Silkworm Inc. - About us' (with a description of the company's mission). A footer at the bottom includes copyright information: 'Copyright © Silkworm Inc.' and language selection options: 'English' and '+ Add a language...'. The top navigation bar includes 'WEBSITE', 'Pages', 'Customize', 'Promote', 'Unpublished' (which is red), and 'Edit'.

When you click the red **Unpublished** slider, the button will change to green and should now display **Published**, as shown in the following screenshot:



Your product has now been published and should be visible to anyone who accesses your website. By clicking the button again, you can set the product back to the **Unpublished** state.

Additional configuration options for your Odoo shop

When you first install Odoo eCommerce, it provides you with a default layout to get you started. However, there are several options available that you can use to alternate how your shop looks and what kind of information it displays. Let's take a look at a few of those options now.

You will want to choose **Shop** from the main menu in order to see your changes. By clicking on the **Customize** menu on the left, you will get a list of options that can be checked or unchecked to change the appearance of your online store.



The **Customize** menu is available on every page, but the options that are available in the menu change depending on what page you are on.

Let's check the **Add to Cart** option to add a small image to each product that will allow visitors to add the product to the shopping cart as shown in the following screenshot:

A screenshot of the Odoo Shop interface. The top navigation bar shows 'Shop | My Website' and 'localhost:8069/shop#scrollTop=0'. Below it is a toolbar with 'WEBSITE', 'Pages', 'Customize', 'Promote', 'New', and 'Edit'. The main content area features a header with the logo 'SILKWORM' and navigation links for 'Home', 'Design Tool', 'Frequently Asked Questions', 'Contact us', 'Shop', and 'Administrator'. A search bar and a 'Sort by' dropdown are also present. The main content displays a grid of products:

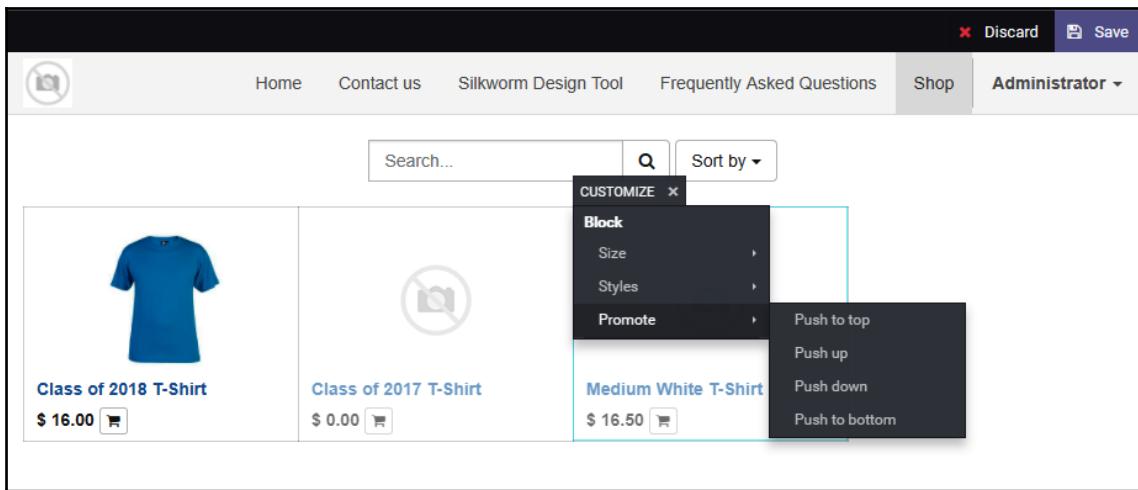
Class of 2018 T-Shirt \$ 16.00	Medium White T-Shirt Unpublished \$ 16.50	Bumper Sticker Unpublished \$ 1.00	Bumper Sticker Unpublished \$ 1.00
Hat Unpublished \$ 1.00	Fan Appreciation Bundle Unpublished \$ 45.00	Class of 2017 T-Shirt Unpublished \$ 25.00	Coffee Mug Unpublished \$ 22.00
Sample Product Unpublished \$ 1.00			

In the previous screenshot, you can see the available list of options, as well as the shopping cart icon to the right of the price of each product. Feel free to experiment with the other options that are available inside the **Customize** windows to get the appearance you want.

Modifying the order of the products on the store

There are also other store options available for you so that you can edit the shop page. One of them is the ability to reorder products. Click **Edit** at the top of the page and then hover over products to see the options available.

We will choose **Promote** and then **Push to top** to send the **Medium White T-shirt** to the top of the page, as shown in the following screenshot:

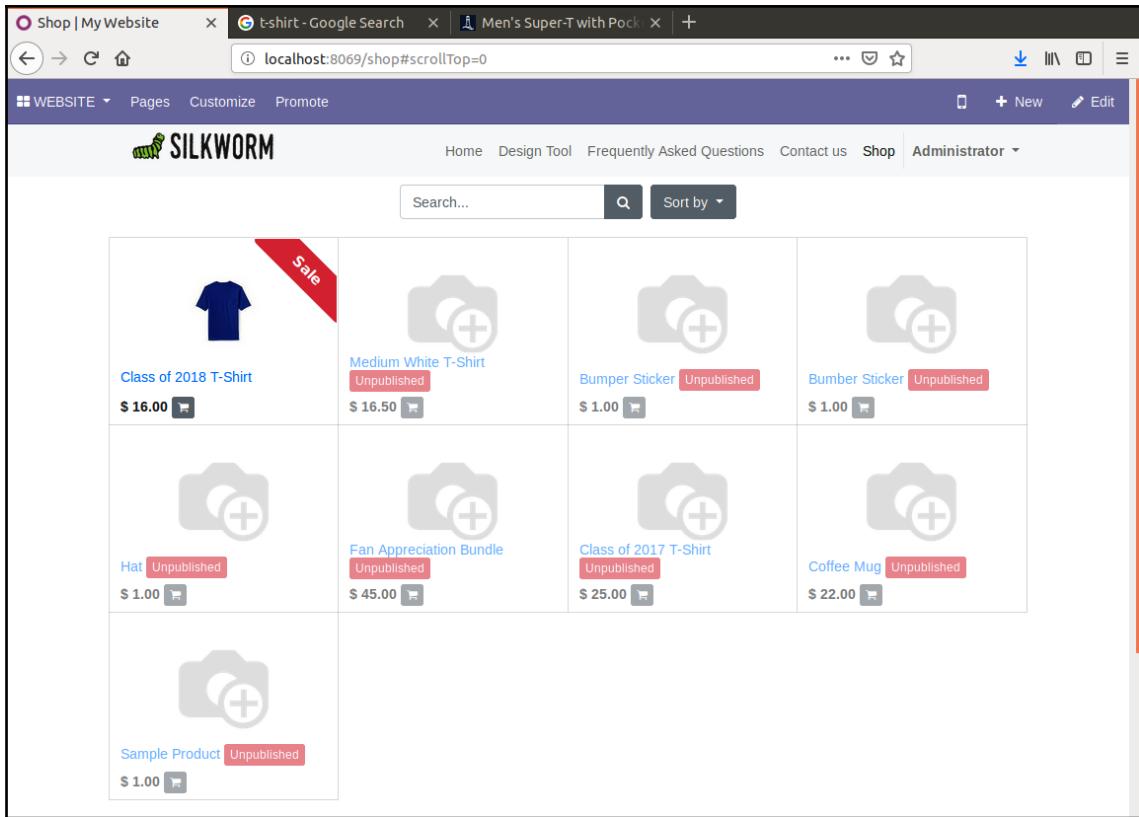


As you can see, it is possible to promote a product to the top of the page, but it could be time-consuming if you decide to move a lot of products around.



Inside Odoo 8's **Product** page, you could quickly adjust the order of the products by setting sequence numbers directly. Unfortunately, this has been removed in Odoo 12. By using the skills that you will learn in Chapter 14, *Modifying Documents and Reports*, on customizing Odoo, you will be able to add the sort order field to the product view.

Take a few minutes to explore the other options in the menu. The size options allow you to change the size of the product when it shows up in the grid, giving you the opportunity to make more interesting shop layouts. The **Styles** option allows you to add a **Sale** ribbon to the product, as you can see in the following screenshot:

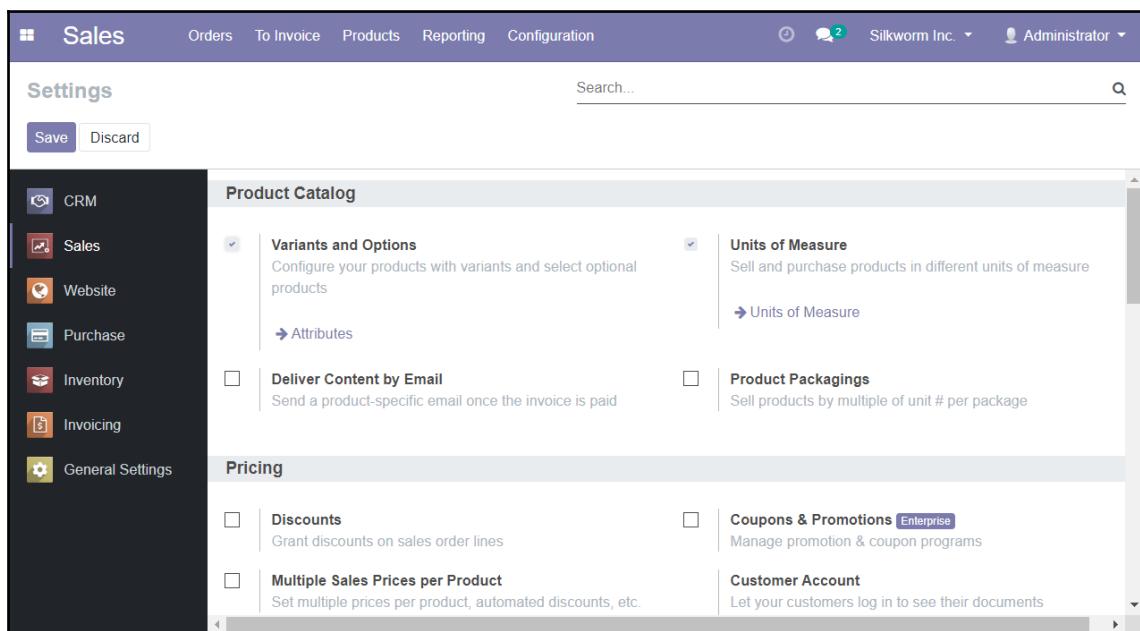


As you can see, Odoo provides quite a few options so that you can customize the display of your eCommerce store. Make sure that you take the time to try them out and do some testing with your potential customers to find out what works best for your business.

Setting up product variants

Another nice feature that comes with the eCommerce application is the ability to create product variants. This is a particularly important feature for the business case we have been using throughout this book. Product variants allow you to offer different options for the same product; for example, a T-shirt will often come in various sizes and colors, or a computer company may want to offer a product with different memory options.

In Odoo 12, product variants are not turned on by default in the eCommerce application. To turn on the product variants option, click on **Settings** in the **Configuration** section under the **Sales** menu:



Under **Product Catalog**, select the **Attribute and Variants** option.

To create product variants, you need to navigate to the product you wish to create the variants for. Let's create variants for **Class of 2018 T-shirt**.

After you have exited the edit mode in the Website Builder, choose **Sales** from the **Website** menu in the top-left corner of your screen. Then, choose products and bring up the **Class of 2018 T-shirt**.

Choose the **Variants** page, and you will see the grid where we will add the variants:

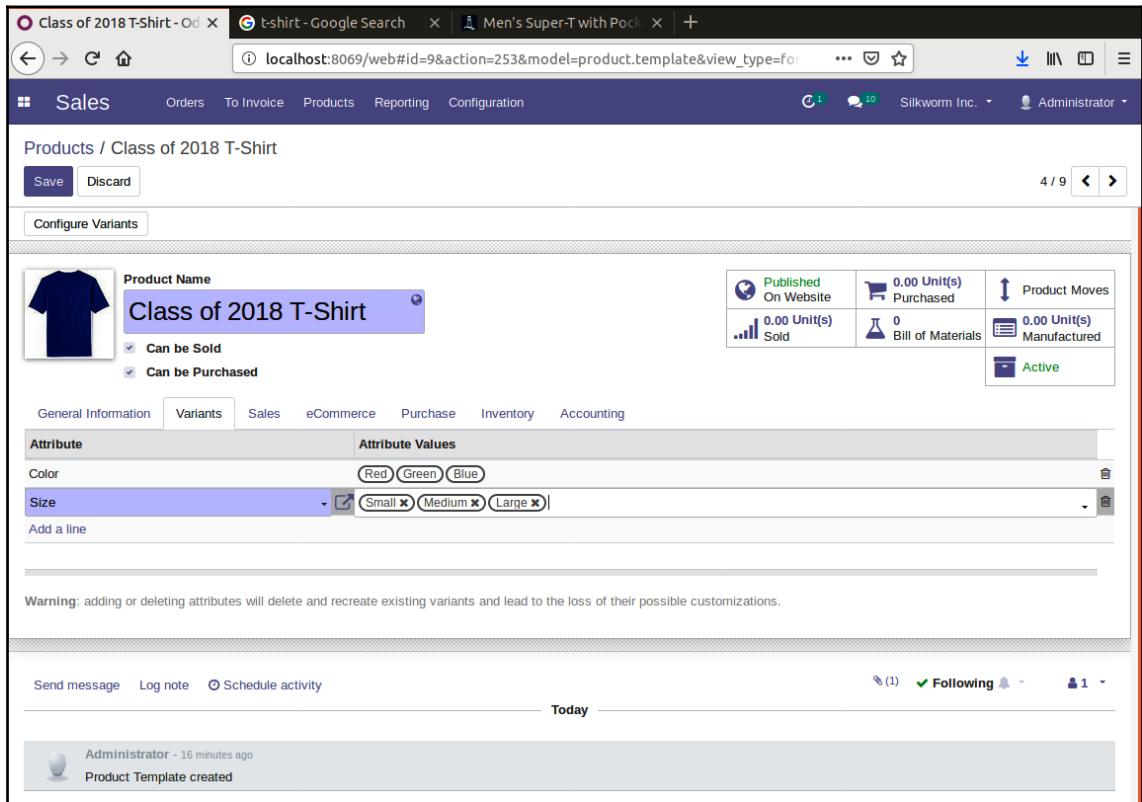
The screenshot shows the Odoo website builder interface for managing product variants. At the top left is a thumbnail of a blue t-shirt. To its right, the product name "Class of 2018 T-Shirt" is displayed in bold. Below the name are two checked checkboxes: "Can be Sold" and "Can be Purchased". To the right of these checkboxes is a summary box containing four status indicators: "Published On Website" (green), "Product Moves" (green), "\$ 0 Sales" (green), "Purchases" (blue) with a value of 0, "Bill of Materials" (blue) with a value of 0, "Manufacturing" (blue) with a value of 0, and "Active" (green). Below this summary are six tabs: General Information, Variants (which is selected and highlighted in blue), Sales, Purchase, Inventory, Invoicing, and Notes. Under the Variants tab, there is a table with two columns: "Attribute" and "Attribute Values". The "Attribute" column is currently empty, while the "Attribute Values" column contains three empty rows. A warning message at the bottom states: "Warning: adding or deleting attributes will delete and recreate existing variants and lead to the loss of their possible customizations."



Note on the right that there is a separate button that brings up a list of variants with the number 1 in the button. You always have one variant, which is the base variant of the product.

Now, click the **Edit** button at the top of the page and add some variants to the product.

To add variants, simply edit the product, choose the **Variants** tab, and click **Add an Item**. In the **Attribute** column, you provide the label you want to use for your attributes. We will create attributes for both **Color** and **Size**. In the product's **Attribute Values** column, you add each of the available values. Here, we will list the available sizes and colors for our product as shown in the following figure:



As you can see in the previous screenshot, we have created two attribute categories, one for **Color** and one for **Size**. For the color attribute, we have specified red, green, and blue. For the size attribute, we have specified **Small**, **Medium**, and **Large**.

Save the product and then go back to the website shop and see the options as they now appear on the **Products** page as shown below:

The screenshot shows a web browser window with three tabs: 'Class of 2018 T-Shirt | My ...', 't-shirt - Google Search', and 'Men's Super-T with Pock ...'. The main content area is a product page for a 'Class of 2018 T-Shirt' from 'SILKWORM'. The page includes a navigation bar with 'WEBSITE', 'Pages', 'Customize', 'Promote', 'Published', '+ New', 'Edit', and 'Administrator'. Below the navigation is a breadcrumb trail 'Products / Class of 2018 T-Shirt' and a search bar. The product title is 'Class of 2018 T-Shirt' with a blue t-shirt image. It has color options (red, green, blue) and size options (Small, Medium, Large). The price is \$16.00 with a quantity selector set to 1. A large blue 'Add to Cart' button is present. Below the product details, there is a guarantee message: '30-day money-back guarantee', 'Free Shipping in U.S.', and 'Buy now, get in 2 days'. At the bottom, there are links for 'Our Products & Services' (Home), 'Connect with us' (Contact us), and 'Silkworm Inc. - About us' (About us). The footer notes: 'We are a team of passionate people whose'.

As you can see, Odoo can take your product attributes and display them in a way that makes it easy for your customers to get the products they wish.

Advanced eCommerce product options

In addition to the basic options we have covered so far, Odoo offers several advanced options that give you more control over how products appear on the store. These options include specifying multiple categories, alternative products, and accessories, as well as direct control over the position of the item on the store.

Once again, you will need to go to the **Products** page and observe that we have specified a few advanced options:

The screenshot shows the Odoo Product Template page for a product named "Class of 2018 T-Shirt".

Product Information:

- Image:** A dark blue t-shirt.
- Product Name:** Class of 2018 T-Shirt
- Status:** Published On Website
- Product Type:** Product
- Checkboxes:** Can be Sold (checked), Can be Purchased (checked)

Tab Navigation: General Information, Variants, Sales, **eCommerce**, Purchase, Inventory, Accounting

Shop Section:

- Categories:** T-Shirts, School, 2018
- Alternative Products:** Class of 2017 T-Shirt
- Accessory Products:** Sport Logo

Extra Images: Add button

In the preceding screenshot, instead of our product now showing up inside of one category, it will show up in the **T-shirts**, **School**, and **2018** categories. This gives us more flexibility when we're designing our eCommerce store.



To see the categories on your website, you will need to go under the **Customize** tab on the store page and check the **Product Categories** option.

Alternative products

Often, when customers purchase one product, it's likely there are alternative products that they may also want to view for comparison. Perhaps it is another brand of product, or a deluxe version of the model they are currently viewing. For our example, we have chosen the medium T-shirt as a potential alternative.

Now, when the user goes to the product page, they will see the alternative product displayed at the bottom but only if it is marked as Published.

Accessory products

These are products that may accompany or complement a product purchase. For our business example, we have chosen a sports logo that can be bought as a pin or a logo on the shirt. A more standard eCommerce example would be if you purchased a tablet computer; you would be offered a case, a stylus, or perhaps even a warranty.

The accessory products are presented when a product is added to the cart, but only if these products are also marked as Published.

Looking at the shopping cart

While we have primarily been focusing on configuration, we also need to remember the website's users. Let's see what the shopping cart looks like for them by adding the **Class of 2018 T-shirt** to a shopping cart, as shown in the following screenshot:

The screenshot shows the Odoo e-commerce interface. At the top, there are navigation links: WEBSITE, Pages, Customize, Promote, Home, Shop, Contact us, My Cart (with 1 item), and Administrator. The main content area has three tabs: Review Order, Address, and Confirm Order. The Review Order tab is active, showing a product: Class of 2018 T-Shirt (Quantity: 1, Price: \$ 16.00). Below it is a section for Suggested Accessories: Sport Logo (\$ 1.00) with an Add to Cart button. At the bottom of this section are Continue Shopping and Process Checkout buttons. The Address tab shows a progress bar with three steps: Step 1 (Review Order) is completed (blue), Step 2 (Address) is in progress (orange), and Step 3 (Confirm Order) is not yet started (grey). The Confirm Order tab is shown on the right, containing an Order Total summary: Subtotal: \$ 16.00, Taxes: \$ 2.40, Total: \$ 18.40. It also includes an I have a promo code input field and a Process Checkout button. At the bottom of the page, there are sections for Our Products & Services (Home), Connect with us (Contact us, phone, email icons), and Silkworm Inc. - About us, which describes the company as a team of passionate people improving lives through products.

You will notice that the **Sports Logo** has been added as a suggested product to our shopping cart, along with a link to add it to the cart. Odoo's shopping cart works like most eCommerce shopping carts do. After you adjust your quantity and are finished shopping, you will click **Process Checkout**, which will take you to the form that collects your shipping and billing information. A progress bar at the top right of the form keeps the user informed of the steps in the process.



Because we are logged in as an administrator, we will see different items on the screen than if we were a guest who was using the shopping cart. One trick is to open an incognito window in our browser to see the exact look and feel for a potential buyer.

Seeing the draft sales order in Odoo

As soon as an item is added to the shopping cart, the order will appear as a draft within Odoo in real time. To see this, go into **Sales** and look in the list of quotes, where you will see your eCommerce order listed at the very top. Click on it to see the details of the order so far:

The screenshot shows the Odoo Sales Quotations interface. At the top, there are three tabs: 'SILK-SO5201 - Odoo', 't-shirt - Google Search', and 'Men's Super-T with Pock'. The address bar shows 'localhost:8069/web#id=5&action=257&model=sale.order&view_type=form&me...'. The main navigation bar includes 'Sales', 'Orders', 'To Invoice', 'Products', 'Reporting', and 'Configuration'. On the right, there are icons for 'Administrator' and other user information.

The main content area is titled 'Quotations / SILK-SO5201'. It has buttons for 'Edit' (highlighted), 'Create', 'Print', 'Action', 'Send by Email', 'Print', 'Confirm', 'Preview', and 'Cancel'. A progress bar indicates the status: 'Quotation' (blue), 'Quotation Sent' (green), and 'Sales Order' (grey). Below this, the order number 'SILK-SO5201' is displayed.

The order details section shows the following fields:

Customer	Administrator	Validity	
Invoice Address	Administrator	Pricelist	Public Pricelist (USD)
Delivery Address	Administrator	Payment Terms	Immediate Payment

The order lines section contains one item:

Product	Description	Ordered Qty	Unit of Measure	Unit Price	Taxes	Subtotal
Class of 2018 T-Shirt (Red, Small)	Class of 2018 T-Shirt (Red, Small)	1.000	Unit(s)	16.00	(Tax 15.00%)	\$ 16.00

At the bottom, there are buttons for 'Send message', 'Log note', and 'Schedule activity'. On the right, there are 'Follow' and '1' notifications.

If the user abandons the order without checking out, it simply remains in draft mode and will be available for deletion later. This is a good way of seeing how many of your users have added something to a cart but didn't make it all the way to the checkout.



Even if a user doesn't check out, this information can be valuable to see what customers are looking for.

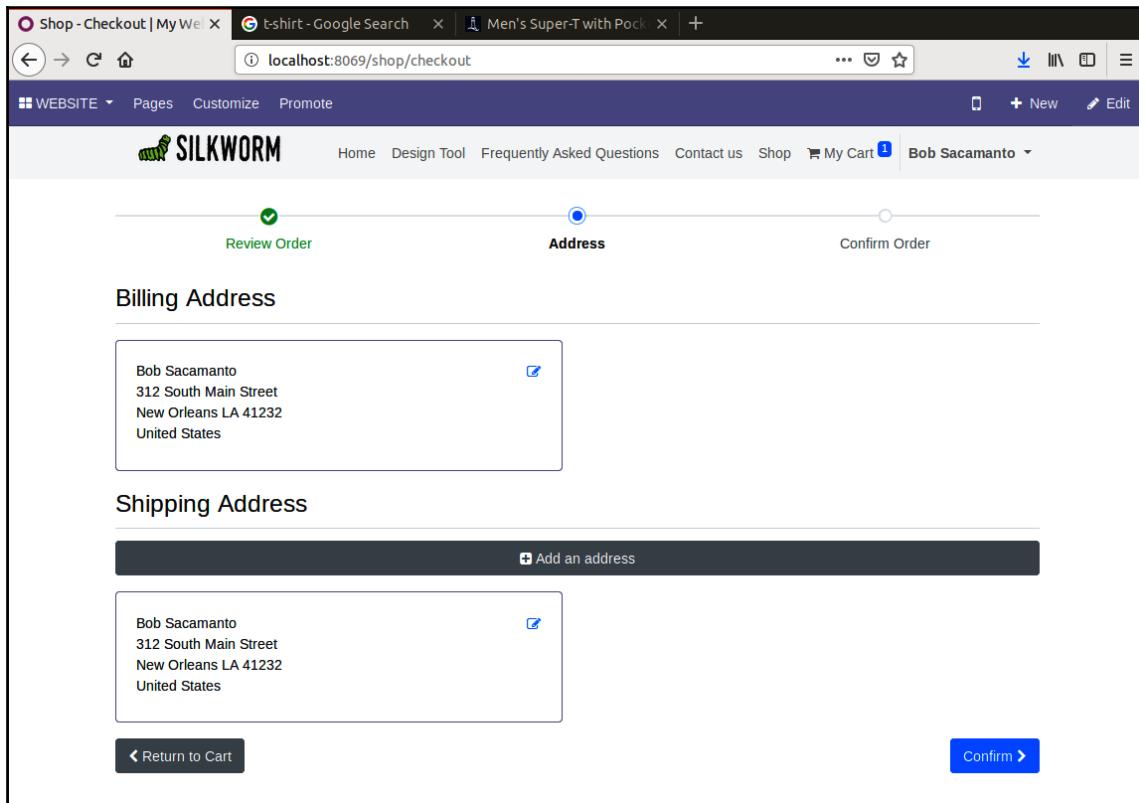
Checking out

Providing a billing address and shipping address is an essential part of the checkout process. Odoo, by default, assumes that the **shipping address** is the same as the **billing address**:

The screenshot shows a web browser window with the URL `localhost:8069/shop/address?partner_id=3`. The page title is "Address Management". The main content area is titled "Your Address". A yellow callout box contains the text: "Be aware! You are editing your **billing** and **shipping** addresses at the same time! If you want to modify your shipping address, create a [new address](#)." Below this, there are fields for Name (Bob Sacamanto), Email (bob@vandalayindustries.com), Phone (111-222-3333), Street and Number (312 South Main Street), Street 2 (Street 2), City (New Orleans), Zip Code (41232), Country (United States), and State / Province (Louisiana). At the bottom are "Back" and "Next >" buttons.

In Odoo 12, all of the fields on this form (in bold) are required except for the **Zip Code** and **Street 2** fields.

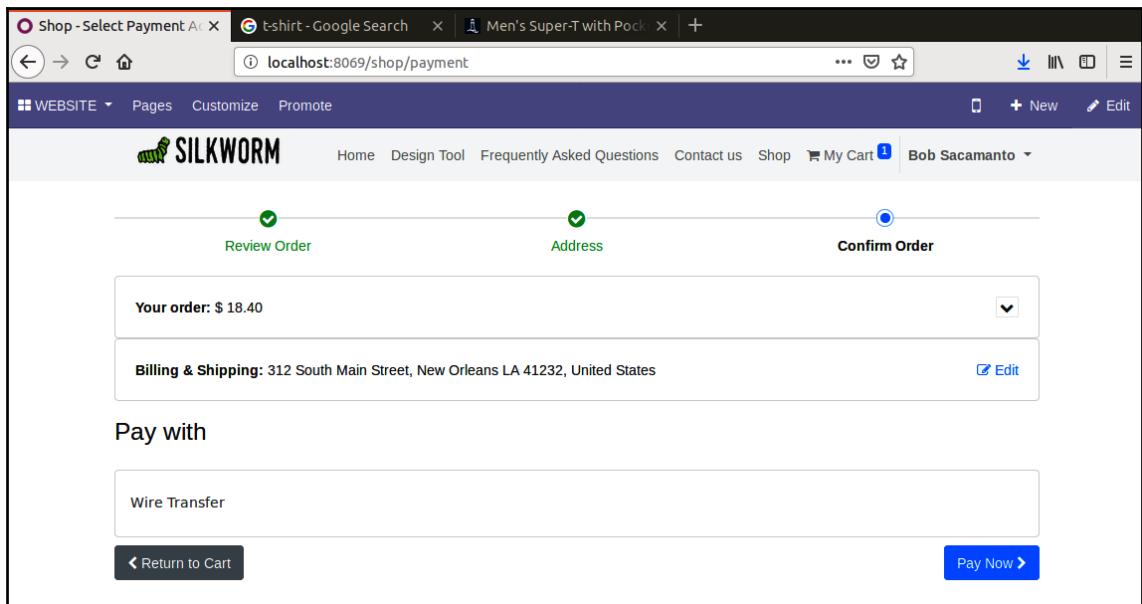
After clicking **Next**, you will be prompted to verify your address information, as shown in the following screenshot:



Here, you are presented with both the **Billing Address** and **Shipping Address** fields, which will both be the same initially. If you use the small icon in the upper-right corner of the address box to edit the address, those edits will apply to both the billing and shipping address. To add an alternate shipping address that is different to their billing address, the customer will need to click **Add an address**.

As customers are using these forms, they are, for the most part, easy to understand. Furthermore, Odoo automatically creates a customer record. All the work is done by the customer! It is important, however, for you to familiarize yourself with the checkout process as it is up to your company to make sure it works as you wish. Depending on your sales volume and the nature of your business, it may be necessary to customize the checkout process to maximize your sales.

Once a user clicks **Confirm**, the onscreen status changes to **Payment**, and they are taken to a page titled **Confirm Order**. By default, Odoo installs **Wire Transfer** as the one and only method of payment:



Now that we have reached this point, the remainder of the process is just like any other eCommerce system. You will notice, however, that the only payment method that is available is **Wire Transfer**. Next, we will see how we can add PayPal as a payment processor.

Adding PayPal as a payment processor

While Odoo only includes **Wire Transfer** by default, the framework is modular and can be extended to include additional payment methods.



One of the most popular eCommerce payment processors, PayPal, can be quickly installed and integrated into your Odoo eCommerce website.

We have to install a PayPal payment processor a little differently because it isn't a full application; instead, it's a module. We still need to go to **Settings** and **Local Apps**, but we will need to uncheck the **Apps** filter from the search.

After clearing the **Apps** filter, simply type in `PayPal` to see the standard **Install** button:

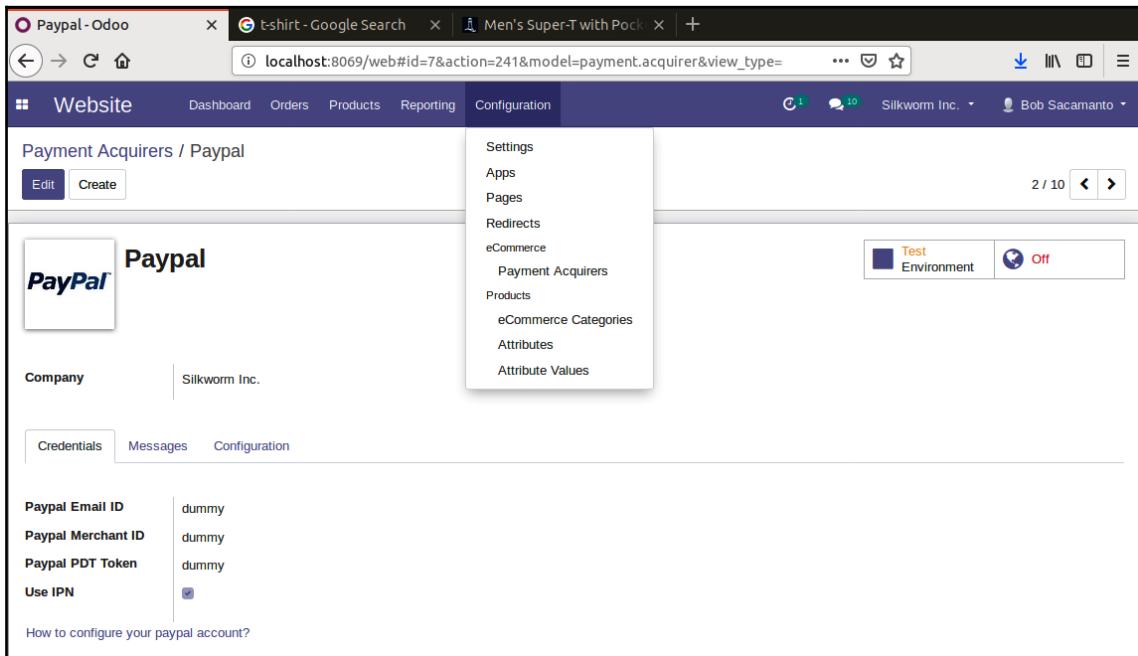
A screenshot of a web browser window showing the Odoo Apps interface. The URL in the address bar is `localhost:8069/web#action=32&model=ir.module.module&view_type=kanban`. The main content area shows a list of modules. One module is highlighted: "Paypal Payment Acquirer" with the sub-label "Payment Acquirer: Paypal Implementation". Below the module name are two buttons: "Install" (which is highlighted in blue) and "Module Info". The top navigation bar shows tabs for "Apps - Odoo", "t-shirt - Google Search", and "Men's Super-T with Pock". The top right corner shows user information for "Bob Sacamano".

Once you've clicked **Install**, the screen will refresh and you will be left on the **Apps** screen with the apps filter back in place. Now, we need to go to the **Website Admin** menu and choose the **Payment Acquirers** option in the **eCommerce** section to bring up the list of payment acquirers. You should now see the **PayPal** provider in that list.



As you can see, it is possible to install a **Paypal Payment Acquirer** through this list rather than having to go into the **Apps** form and installing it there.

Click **PayPal** to open the form and view the available options, as shown in the following screenshot:



Under the **Environment** option, you should see that Odoo defaults to the **Test** environment, which PayPal calls the sandbox. This allows you to configure your **Paypal Email ID** and **PayPal Merchant ID** at the bottom and begin testing your store. Once you have everything worked out, you can turn the environment from **Test** to **Production**.

Naturally, you will need to set up a PayPal account and use the credentials they give you to fill out the form.

Summary

In this chapter, we learned how Odoo eCommerce fits in with the Website Builder, how to install it, and the basic configuration options. Next, we looked at product variants and how they can be used to make it easier to present products that come in multiple styles.

We also spent a little time learning about the advanced product options Odoo offers, such as alternative products, accessories, and defining multiple categories. After looking at the checkout process, we learned how to add an additional payment processor, PayPal, and where you need to go to set the options that are required to make it all work.

In the next chapter, we will look at one of the more exciting aspects of working with Odoo: how you can customize Odoo to meet the specific needs of your organization. We will discover how to activate Developer mode, which will allow us to append fields to Odoo's screens. We'll then begin adding our own fields to forms in order to collect data, to lists in order to find and display data, and to models, which handle all the rules and methods for storing the data of the underlying PostgreSQL database. But first, what is the single most important thing that we need to do before customizing our system? That's right... we need to know how to backup and restore our data, just in case.

13

Customizing Odoo for Your Business

In this chapter, we will cover one of the greatest advantages of Odoo: the ability to customize the software to meet the unique needs of your business. Fortunately, Odoo provides a great deal of flexibility, which allows you to customize Odoo without writing any code or developing any modules. We will begin by learning about how to activate Odoo's developer mode and then back up our database, which is a very important practice when customizing Odoo. After that, we will learn about how to add fields to our database and display them on forms and list views. Note that customizing Odoo is a very broad topic that would ideally span across multiple chapters; you might consider this as an introduction to customization.

The following topics will be covered in this chapter:

- Understanding the Odoo architecture
- Entering and exiting developer mode
- Backing up your database
- Restoring data from a backup
- Appending custom fields to models
- Displaying newly added fields in forms and list views
- Using Odoo's API to integrate with other applications

Understanding the Odoo architecture

Before you can begin to understand Odoo, it is important that you have a basic understanding of the framework and underlying architecture that makes up Odoo applications. Each Odoo application has three primary components that make up the final Odoo application. Fortunately, you don't have to be an expert developer to understand how the Odoo framework fits together. The three components are **models**, **views**, and **actions**.

Models

In the Odoo framework, models are what hold and manage the data that makes up your Odoo application. When you save a sales order in Odoo, the data for the sales order header is stored inside the `sale.order` model. Individual data items, such as order dates and customer addresses, are known as **fields**.

Models can also be linked and associated with other models. For example, the `sale.order` model is linked to the `sale.order.line` model by the ID of the sales order. In this chapter, we will be adding a few fields to the `sale.order` model via Odoo.

Views

Models by themselves do not display any information to the end user. Models simply hold and manage the data for the applications behind the scenes. Data that is collected from these models is displayed in your applications using views. This allows the `sale.order` model information to be displayed in a variety of different ways. Do you want to make a very simple custom order screen that summarizes orders for the day? Create a custom view to show the model information in whichever way you wish.

Actions

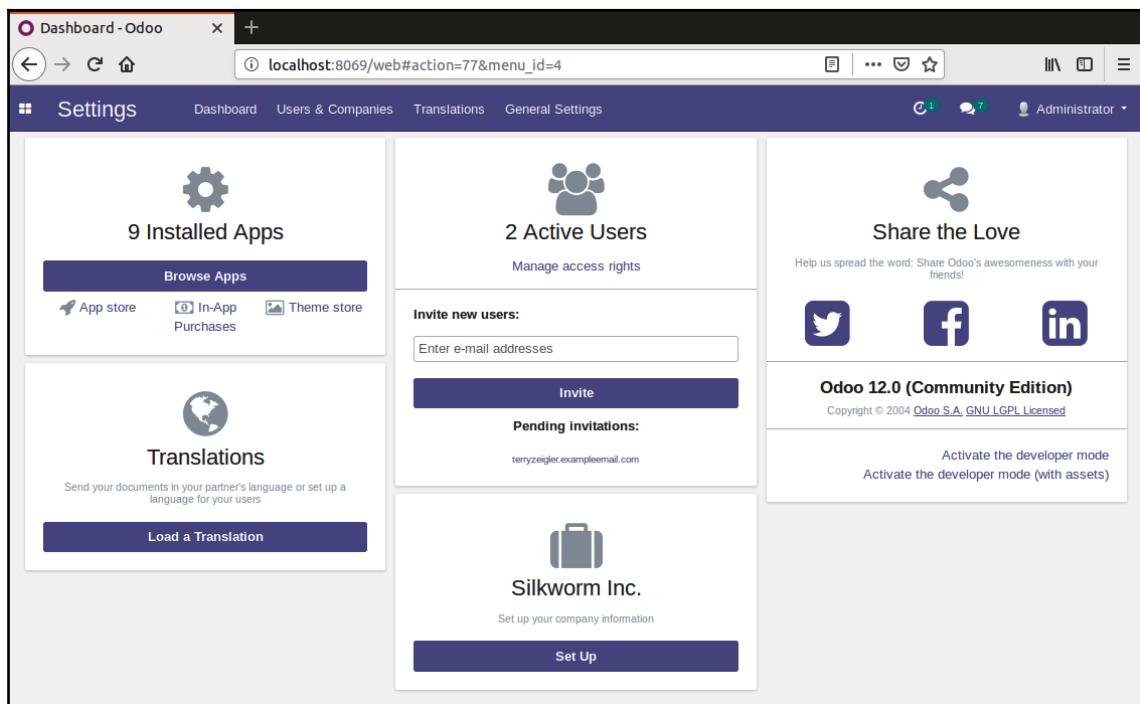
The third and final component that pulls together the framework is actions. Actions are what trigger the appropriate views to be displayed, or for specific actions to take place on a model. For example, when you choose **Quotations** from the **Sale** menu, you are triggering an action that tells the Odoo framework to display the appropriate `sale.order` view.

Without actions, the Odoo framework would not know which views to display. Another example of an action would be posting or confirming a sales order. When you click **Confirm**, the Odoo framework calls the appropriate function that then updates the `sale.order` model.

Activating developer mode

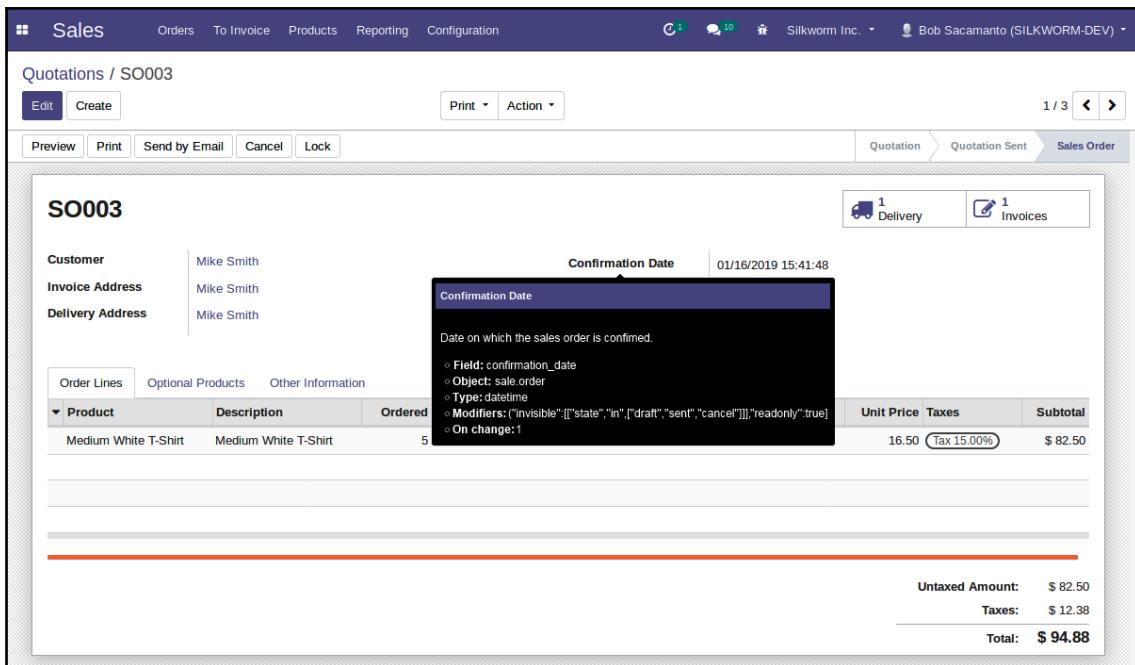
To customize Odoo, you must first activate developer mode. Once you enter this mode, Odoo will provide you with a lot more onscreen information as you navigate through the interface. This mode also allows you to make changes to the database and store that information in a file.

To activate developer mode, click on the **Settings** menu. On the far right, you will see a panel that contains information about the Odoo installation. At the bottom of the panel, you will see the link to **Activate the developer mode**:



Once you have entered this screen, you can click on **Activate the developer mode** to begin customizing Odoo.

Odoo recognizes that you are in developer mode by adding ?debug=# to the URL in your web browser. Additionally, Odoo changes the information that is provided when you hover over various fields in the interface. For example, when viewing a sales order record while in developer mode, you can move the cursor over the **Confirmation Date** field to reveal details about how that field is represented internally in Odoo, as shown in the following screenshot:

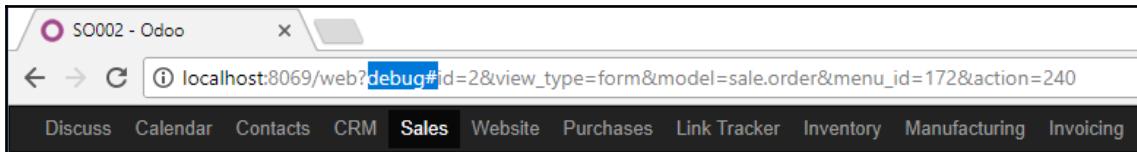


The preceding screenshot demonstrates how Odoo displays information while you are in developer mode. In this example, we can see that the **Confirmation Date** field is named `confirmation_date` and that the field belongs to the `sale.order` object. Additionally, we can see that the field type is `date` and that there are modifiers assigned to the field. This type of information will be of great value as you continue to customize Odoo.

Getting out of developer mode

Once you are in developer mode, there will come a time when you want to exit that mode and continue to work with Odoo as you normally would. To exit developer mode, simply go back to the **Settings** menu and choose the **Deactivate the developer mode** link.

Alternatively, you can simply remove `debug#` from the URL string in your browser:



Make sure you leave the `&` symbol in place when you remove the `debug#` tag from the URL. If you get any errors or other unusual behavior after removing `debug#` from your URL, you can typically use your browser's back button.

If this also fails, you can always restart the browser and log back in to Odoo.

Backing up your database

When you make changes in developer mode, those changes are written into the database associated with the company. One of the major advantages of this approach is that you do not have to write code in Python or create a custom module to implement simple customization. You don't even have to restart the server. One of the major disadvantages, however, is that there is the potential for making an unwanted change, or one that is difficult to reverse.

Therefore, it is very important and highly recommended that you make backups of your database both before and after you make any customization while in developer mode.

Do not skip this step!



It is always a good idea to frequently back up your database, but it is absolutely imperative that you back up your database before undertaking any customization.

To back up your database, you must first log out of Odoo. After you have successfully logged out, click on the **Manage Databases** link on the login form. From this screen, you click on the **Backup** link of the **Database Management** menu as shown in the following screenshot:



To back up your database, select the database from the pop-up menu and enter the master password for the Odoo installation (by default, the master password will be `admin`). Next, click on the **Backup** button.

After you click on the **Backup** button, Odoo will then save your database to your local drive. Depending on the browser you are using and its settings, the prompt you get to save your file will vary. The default filename will end with the `.dump` extension.

After you save your file, Odoo will download it into the directory you have specified. If this is the first time you have backed up your database, you should also take the time to verify that you can successfully restore the database. While this may seem like an unnecessary exercise, it is important to remember that a backup is only as good as your ability to successfully restore it.

Restoring a database in Odoo

To restore a database in Odoo, click on the **Restore** option in the **Database Management** menu.

To restore your database, you need to provide three pieces of information: the backup file you wish to restore, the master password for Odoo, and a new database name. Clicking on the **Browse...** button in the file selection area will prompt you to select the `.dump` file that was created when you performed the backup.

After you have specified the file and the other required fields, click on **Restore** to begin restoring the database. A small progress bar in the bottom left of the browser will update you on the progress of the restore. Once the restore is complete, log in to the database to make sure everything is working as expected.

Now with a successful backup and restoration, you are ready to begin customizing Odoo. If something goes wrong, you will now have the ability to restore your backup. While customizing Odoo, remember to back up the database frequently.

Adding a custom field to Odoo

One of the most common reasons for customizing Odoo is to collect additional information that is specific to your company. If you are running an insurance company, for example, you may want to specify the policy number on your sales order. If you are working in property management, you might want to store the date on which the lease agreement will expire.

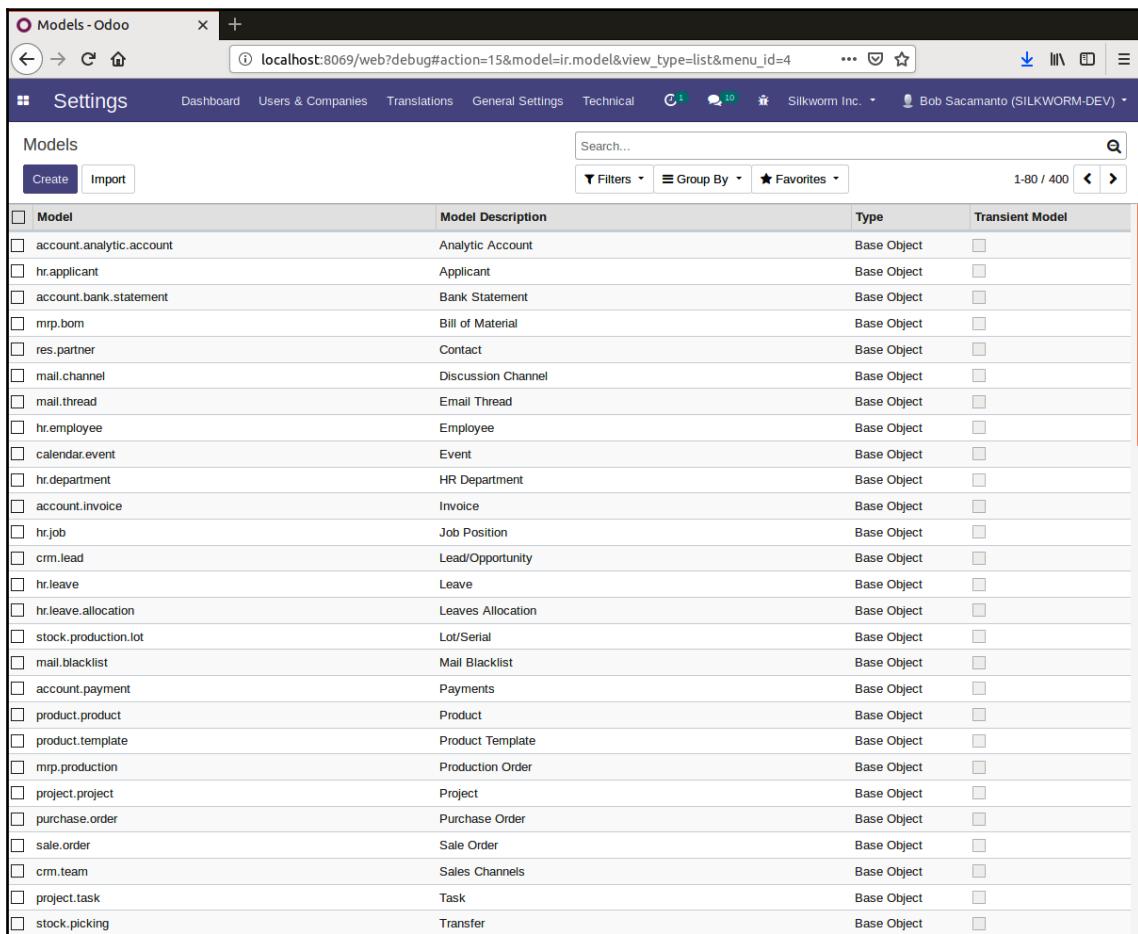
For our working example, we will be adding fields that will help us better manage the data and processes for our silkscreen company. Specifically, we will be adding the following fields to the sales order header:

Field name	Label	Field type	Purpose
<code>x_daterequired</code>	Date Required	Date	In the screen-printing industry, deadlines drive when production begins and when the product should be delivered to the customer.
<code>x_rush</code>	Rush Order	Boolean	Related to Date Required is the necessity to flag some sales orders as rush orders. A Rush Order can then be prioritized and given expedited treatment.

Custom field names in Odoo are preceded by `x_`. This is so that field names in future Odoo versions and standard updates will not accidentally conflict with the custom fields you have added.

Viewing the model in Odoo

Odoo allows you to add custom fields to the model if you are in debug mode. Click on **Settings** and then, in the **Database Structure** sub-menu, choose **Models**. You will get a list of all the models that make up your current Odoo installation, as shown in the following screenshot:



The screenshot shows the Odoo web interface with the title "Models - Odoo". The URL in the address bar is "localhost:8069/web?debug#action=15&model=ir.model&view_type=list&menu_id=4". The page is titled "Settings" and shows a list of "Models". There are two buttons at the top left: "Create" and "Import". A search bar and filter options are at the top right. The main table has columns for "Model", "Model Description", "Type", and "Transient Model". The table lists various Odoo models such as account.analytic.account, hr.applicant, account.bank.statement, mrp.bom, res.partner, mail.channel, mail.thread, hr.employee, calendar.event, hr.department, account.invoice, hr.job, crm.lead, hr.leave, hr.leave.allocation, stock.production.lot, mail.blacklist, account.payment, product.product, product.template, mrp.production, project.project, purchase.order, sale.order, crm.team, project.task, and stock.picking.

Model	Model Description	Type	Transient Model
account.analytic.account	Analytic Account	Base Object	
hr.applicant	Applicant	Base Object	
account.bank.statement	Bank Statement	Base Object	
mrp.bom	Bill of Material	Base Object	
res.partner	Contact	Base Object	
mail.channel	Discussion Channel	Base Object	
mail.thread	Email Thread	Base Object	
hr.employee	Employee	Base Object	
calendar.event	Event	Base Object	
hr.department	HR Department	Base Object	
account.invoice	Invoice	Base Object	
hr.job	Job Position	Base Object	
crm.lead	Lead/Opportunity	Base Object	
hr.leave	Leave	Base Object	
hr.leave.allocation	Leaves Allocation	Base Object	
stock.production.lot	Lot/Serial	Base Object	
mail.blacklist	Mail Blacklist	Base Object	
account.payment	Payments	Base Object	
product.product	Product	Base Object	
product.template	Product Template	Base Object	
mrp.production	Production Order	Base Object	
project.project	Project	Base Object	
purchase.order	Purchase Order	Base Object	
sale.order	Sale Order	Base Object	
crm.team	Sales Channels	Base Object	
project.task	Task	Base Object	
stock.picking	Transfer	Base Object	

Using the standard search tools in Odoo that you have learned about so far, you can now limit the results to just show the `sale.order` model:

The screenshot shows the Odoo Settings interface with the 'Models' tab selected. The search bar at the top right is set to 'sale.order'. The main list displays various Odoo models with their descriptions and types:

Model	Model Description	Type	Transient Model
sale.order	Sale Order	Base Object	<input type="checkbox"/>
project.create.sale.order.line	Create SO Line from project	Base Object	<input checked="" type="checkbox"/>
project.create.sale.order	Create SO from project	Base Object	<input checked="" type="checkbox"/>
sale.order.template	Quotation Template	Base Object	<input type="checkbox"/>
sale.order.template.line	Quotation Template Line	Base Object	<input type="checkbox"/>
sale.order.template.option	Quotation Template Option	Base Object	<input type="checkbox"/>
sale.order.option	Sale Options	Base Object	<input type="checkbox"/>
sale.order.line	Sales Order Line	Base Object	<input type="checkbox"/>

You can now click the `sale.order` model to open it up and display all the fields that make up the `sale.order` model:

The screenshot shows the Odoo Settings interface with the 'Models / Sale Order' tab selected. The model description is 'Sale Order' and the type is 'In Apps'. The list of fields includes:

Field Name	Field Label	Field Type	Required	Readonly	Indexed	Type
__last_update	Last Modified on	datetime	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
access_token	Security Token	char	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
access_url	Portal Access URL	char	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
access_warning	Access warning	text	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
activity_date_deadline	Next Activity Deadline	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
activity_ids	Activities	one2many	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
activity_state	Activity State	selection	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
activity_summary	Next Activity Summary	char	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
activity_type_id	Next Activity Type	many2one	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
activity_user_id	Responsible User	many2one	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Base Field
amount_by_group	Tax amount by group	binary	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
amount_tax	Taxes	monetary	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
amount_total	Total	monetary	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
amount_undiscounted	Amount Before Discount	float	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
amount_untaxed	Untaxed Amount	monetary	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
analytic_account_id	Analytic Account	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field
authorized_transaction_ids	Authorized Transactions	many2many	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Base Field

Here, you can examine the name of the field as well as the field type and whether the field is required or not. Some fields are also designated as read-only fields. These fields are often automatically generated or calculated by Odoo.



Examining models in Odoo is a great way to learn more about the structure of the data and how it is organized. This is particularly vital for anyone who wishes to customize Odoo.

Creating a new field in the sale.order model

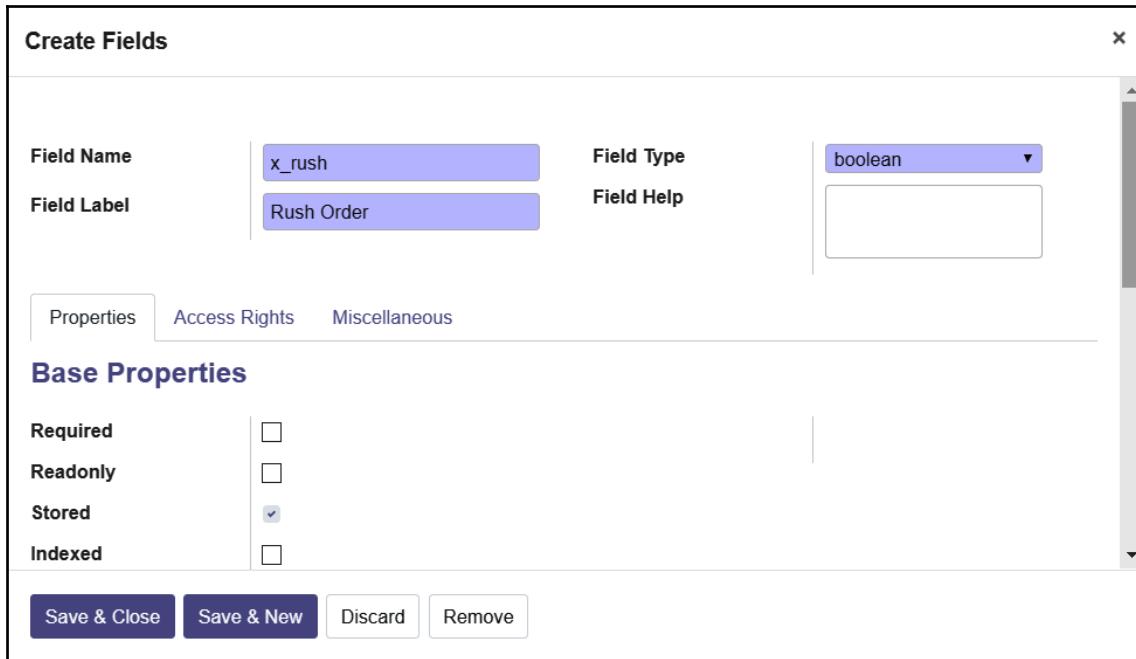
At the top of the form, you can see that the **Model Description** for the `sale.order` model is **Quotation**. For the purposes of this chapter, we will refer to the model, rather than the model description. Let's go ahead and start adding our custom fields to the `sale.order` model. Please be aware that it can be easy to accidentally click **Create** and create an entirely new model. We don't want a new model, but rather want to add fields to the existing model. Click **Edit** to edit the `sale.order` model, and then scroll down to the bottom of the field list and click **Add a Line** as shown in the following screenshot:

A screenshot of the Odoo Model Editor. The window title is "Sale Order - Odoo". The URL bar shows "localhost:8069/web?debug#id=253&action=15&model=ir.model&view_type=form&me". The left sidebar lists various fields like state, tag_ids, tasks_ids, team_id, timesheet, transaction, type_name, user_id, validity_, warehouse, warning, website_id, website, website_, write_date, write_uid, x_date, and x_daterequired. A vertical toolbar on the right has icons for Settings, Dashboard, Users & Companies, Translations, General Settings, Technical, and a search bar. The main area is titled "Open: Fields". It shows a new field being created: "Field Name" is "x_daterequired", "Field Label" is "Date Required", and "Field Type" is "date". Below this are tabs for "Properties", "Access Rights", and "Miscellaneous". Under "Base Properties", "Required" and "Readonly" are unchecked, "Stored" is checked, "Indexed" is unchecked, "Copied" is checked, and "Tracking" is empty. Under "Advanced Properties", "Related Field" and "Dependencies" are empty, and "Compute" is also empty. At the bottom, there are buttons for "Save & Close", "Save & New", "Discard", and "Remove".

In the preceding example, we have specified our **Date Required** custom field.

The `x_` prefix is already specified in the **Field Name** by default to encourage the use of good naming conventions. We have filled in the other data required for the field, including setting the **Field Name** to `x_daterequired`, the **Field Label** to `Date Required`, and the **Field Type** to `date`.

Click on **Save & New** to finish adding a new field to the `sale.order` model, and then proceed to enter the remaining `x_rush` field:

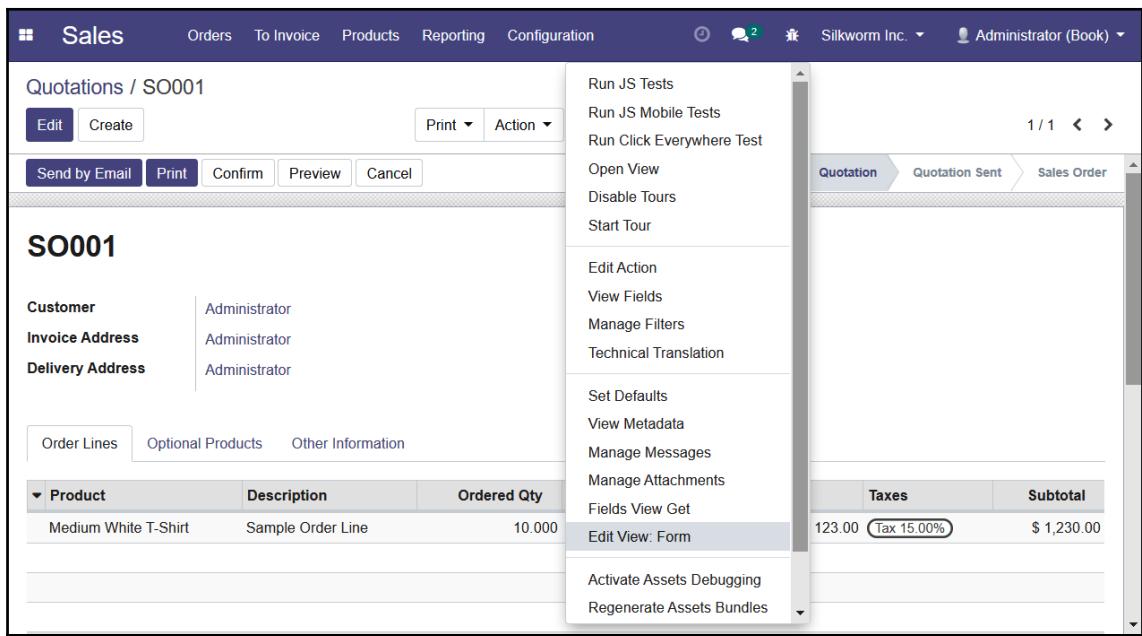


You should notice that this field is very similar to the **Date Required** field, except that we specified the **Field Type** as **boolean**. This will tell Odoo's framework to display the field as a checkbox. Now that we have finished adding our new fields, click on the **Save & Close** button and then hit **Save** on the **Model** screen to finish adding these fields to the `sale.order` model.

Editing the form view

Now that we have added our fields to the model, we want to display them on the form. Fortunately, Odoo provides an editor that makes it easy to add the fields to your view. It would be beneficial to have experience using a text editor with XML here. If you don't, please see the [Appendix A, Locating Additional Odoo Resources](#), where you can find some basic information on editing XML.

The easiest way to edit a view—in this case, the **Sale Order** form—is to go to the form you wish to edit. Simply pull up any sales order in Odoo, and choose **Edit View: Form** from the debug menu as shown in the following screenshot:



This will bring up the actual XML code that makes up the sales order view. It may look somewhat intimidating to newcomers at first, but the changes we are going to make are actually very easy. Even better, you can copy and paste an existing field, so you don't have to type all the special characters and understand the syntax.

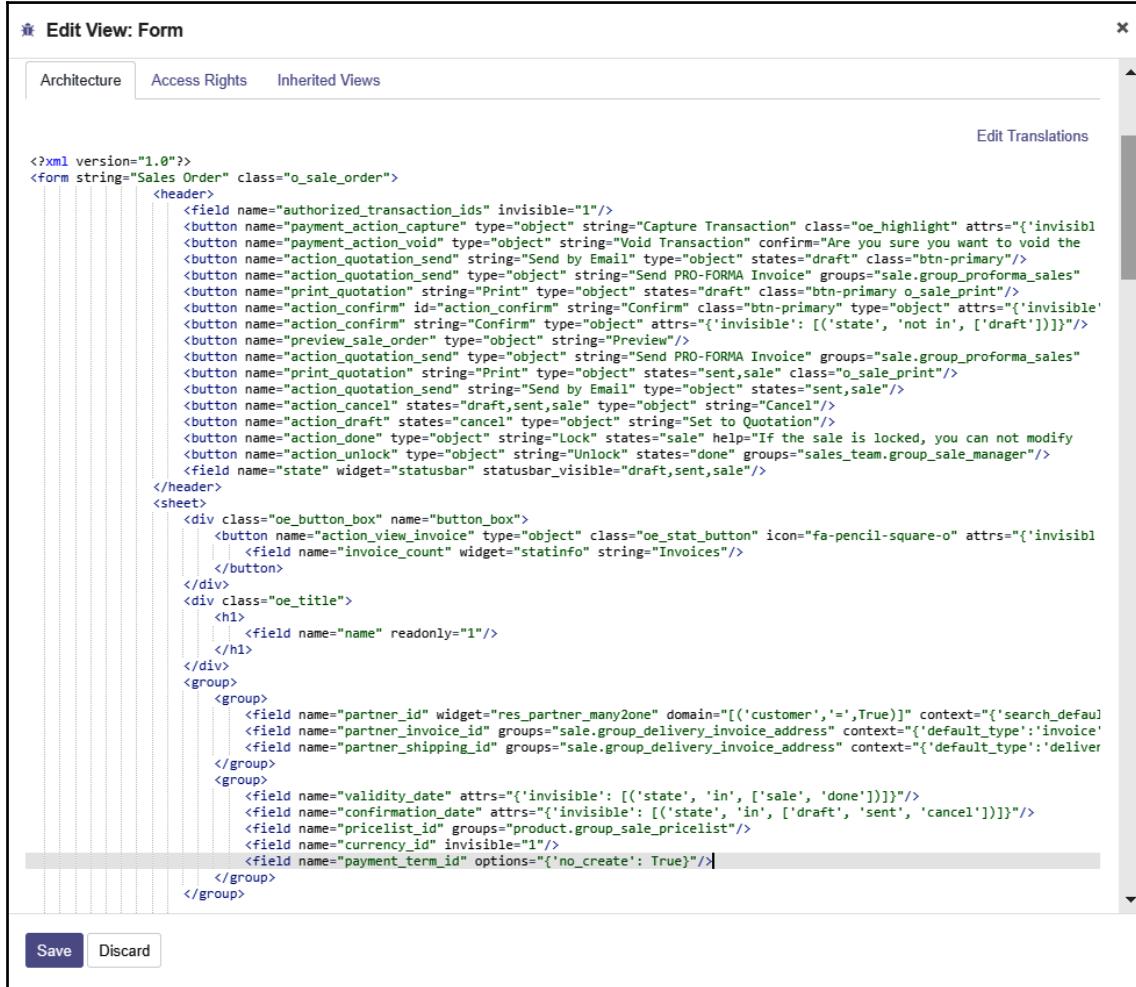
Now, scroll down until you find the line that displays the payment terms on the **Sales Order** form. We will add our two custom fields directly below the line for payment terms, as shown in the following screenshot:

The screenshot shows the Odoo Form View Editor with the XML code for a Sales Order form. The XML code defines the structure of the form, including buttons for various actions like capture, void, and confirmation, and fields for partner information and currency. A new field, 'payment_term_id', is being added to the group section, which contains fields for validity date, confirmation date, pricelist, currency, and payment term.

```
<?xml version="1.0"?>
<form string="Sales Order" class="o_sale_order">
    <header>
        <field name="authorized_transaction_ids" invisible="1"/>
        <button name="payment_action_capture" type="object" string="Capture Transaction" class="oe_highlight" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
        <button name="payment_action_void" type="object" string="Void Transaction" confirm="Are you sure you want to void the transaction?"/>
        <button name="action_quotation_send" string="Send by Email" type="object" states="draft" class="btn-primary"/>
        <button name="action_quotation_send" type="object" string="Send PRO-FORMA Invoice" groups="sale.group_proforma_sales"/>
        <button name="print_quotation" string="Print" type="object" states="draft" class="btn-primary o_sale_print"/>
        <button name="action_confirm" id="action_confirm" string="Confirm" class="btn-primary" type="object" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
        <button name="action_confirm" string="Confirm" type="object" attrs="{'invisible': [('state', 'not in', ['draft'])]}"/>
        <button name="preview_sale_order" type="object" string="Preview"/>
        <button name="action_quotation_send" type="object" string="Send PRO-FORMA Invoice" groups="sale.group_proforma_sales"/>
        <button name="print_quotation" string="Print" type="object" states="sent,sale" class="o_sale_print"/>
        <button name="action_quotation_send" string="Send by Email" type="object" states="sent,sale"/>
        <button name="action_cancel" states="draft,sent,sale" type="object" string="Cancel"/>
        <button name="action_draft" states="cancel" type="object" string="Set to Quotation"/>
        <button name="action_done" type="object" string="Lock" states="sale" help="If the sale is locked, you can not modify it"/>
        <button name="action_unlock" type="object" string="Unlock" states="done" groups="sales_team.group_sale_manager"/>
        <field name="state" widget="statusbar" statusbar_visible="draft,sent,sale"/>
    </header>
    <sheet>
        <div class="oe_button_box" name="button_box">
            <button name="action_view_invoice" type="object" class="oe_stat_button" icon="fa-pencil-square-o" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
            <field name="invoice_count" widget="statinfo" string="Invoices"/>
        </div>
        <div class="oe_title">
            <h1>
                <field name="name" readonly="1"/>
            </h1>
        </div>
        <group>
            <group>
                <field name="partner_id" widget="res_partner_manyZone" domain="['customer']" context="{'search_default_customer': 1}"/>
                <field name="partner_invoice_id" groups="sale.group_delivery_invoice_address" context="{'default_type': 'invoice'}"/>
                <field name="partner_shipping_id" groups="sale.group_delivery_invoice_address" context="{'default_type': 'delivery'}"/>
            </group>
            <group>
                <field name="validity_date" attrs="{'invisible': [('state', 'in', ['sale', 'done'])]}"/>
                <field name="confirmation_date" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
                <field name="pricelist_id" groups="product.group_sale_pricelist"/>
                <field name="currency_id" invisible="1"/>
                <field name="payment_term_id" options="{'no_create': True}"/>
            </group>
        </group>
    </sheet>
</form>
```

Save Discard

To simplify, you can copy and paste the `payment_term_id` line and then edit it to include the custom field you wish to add. Here, we have added the field for **Date Required** to our view:



The screenshot shows the 'Edit View: Form' dialog for a 'Sales Order' model. The XML code defines a form view with various buttons and fields. A specific line has been highlighted with a blue rectangle:

```
<?xml version="1.0"?>
<form string="Sales Order" class="o_sale_order">
    <header>
        <field name="authorized_transaction_ids" invisible="1"/>
        <button name="payment_action_capture" type="object" string="Capture Transaction" class="oe_highlight" attrs="{'invisible': [('state', '=', 'done')]}"/>
        <button name="payment_action_void" type="object" string="Void Transaction" confirm="Are you sure you want to void the transaction?" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
        <button name="action_quotation_send" string="Send by Email" type="object" states="draft" class="btn-primary"/>
        <button name="action_quotation_send" type="object" string="Send PRO-FORMA Invoice" groups="sale.group_proforma_sales"/>
        <button name="print_quotation" string="Print" type="object" states="draft" class="btn-primary o_sale_print"/>
        <button name="action_confirm" id="action_confirm" string="Confirm" class="btn-primary" type="object" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
        <button name="action_confirm" string="Confirm" type="object" attrs="{'invisible': [('state', 'not in', ['draft'])]}"/>
        <button name="preview_sale_order" type="object" string="Preview"/>
        <button name="action_quotation_send" type="object" string="Send PRO-FORMA Invoice" groups="sale.group_proforma_sales"/>
        <button name="print_quotation" string="Print" type="object" states="sent,sale" class="o_sale_print"/>
        <button name="action_quotation_send" string="Send by Email" type="object" states="sent,sale"/>
        <button name="action_cancel" states="draft,sent,sale" type="object" string="Cancel"/>
        <button name="action_draft" states="cancel" type="object" string="Set to Quotation"/>
        <button name="action_done" type="object" string="Lock" states="sale" help="If the sale is locked, you can not modify it."/>
        <button name="action_unlock" type="object" string="Unlock" states="done" groups="sales_team.group_sale_manager"/>
        <field name="state" widget="statusbar" statusbar_visible="draft,sent,sale"/>
    </header>
    <sheet>
        <div class="oe_button_box" name="button_box">
            <button name="action_view_invoice" type="object" class="oe_stat_button" icon="fa-pencil-square-o" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
            <field name="invoice_count" widget="statinfo" string="Invoices"/>
        </div>
        <div class="oe_title">
            <h1>
                <field name="name" readonly="1"/>
            </h1>
        </div>
        <group>
            <group>
                <field name="partner_id" widget="res_partner_manyZone" domain="([('customer', '=', True)])" context="{'search_default_customer': 1}"/>
                <field name="partner_invoice_id" groups="sale.group_delivery_invoice_address" context="{'default_type': 'invoice'}"/>
                <field name="partner_shipping_id" groups="sale.group_delivery_invoice_address" context="{'default_type': 'delivery'}"/>
            </group>
            <group>
                <field name="validity_date" attrs="{'invisible': [('state', 'in', ['sale', 'done'])]}"/>
                <field name="confirmation_date" attrs="{'invisible': [('state', 'in', ['draft', 'sent', 'cancel'])]}"/>
                <field name="pricelist_id" groups="product.group_sale_pricelist"/>
                <field name="currency_id" invisible="1"/>
                <field name="payment_term_id" options="{'no_create': True}"/>
            </group>
        </group>
    </sheet>

```

At the bottom left, there are 'Save' and 'Discard' buttons.

Notice that we have removed the options tag for the purposes of this example. Now, you can save the form and refresh it.



As was explained earlier on in this chapter, changing and modifying Odoo in the database can be dangerous, and can even break your Odoo installation if you make mistakes.

Please be aware that you may have to hold down the *Shift* key when you refresh your browser to see the change. Once properly refreshed, our new field labeled **Date Required** should now appear underneath the **Payment Terms**:

The screenshot shows the Odoo Sales Order interface for Quotation SO003. At the top, there are buttons for Save, Discard, Preview, Print, Send by Email, Cancel, and Lock. Below the header, there are tabs for Quotation, Quotation Sent, and Sales Order. The main area displays the quotation details: Customer (Mike Smith), Confirmation Date (01/16/2019 15:41:48), Pricelist (Public Pricelist (USD)), and Payment Terms (15 Days). A new field, Date Required, is present, showing 04/11/2019. The Order Lines section lists one item: Medium White T-Shirt, ordered quantity 5.000, delivered quantity 0.000, invoiced quantity 5.000, unit price 16.50, tax 15.00%, and subtotal \$ 82.50. The total for the quotation is \$ 94.88, broken down into Untaxed Amount (\$ 82.50), Taxes (\$ 12.38), and Total (\$ 94.88). At the bottom, there are buttons for Send message, Log note, Schedule activity, Following, and a notification count of 3.

As you can see, it is pretty easy to create a new field and then add that field to views and forms. Now, repeat these steps to add our new `x_rush` field to the view:

1. Edit the form view.
2. Copy and paste the `x_daterequired` line of code.
3. Change the field name to `x_rush`.
4. Save your changes.
5. Refresh to view your new field on the form.

With this, we have added the required fields to views and forms.

Customizing search operations in Odoo

In addition to having the ability to modify the forms and list views in Odoo, you can also customize searching in Odoo to better fit the needs of your organization. For example, when customers place orders, it is common in a business-to-business scenario that you will be provided with a purchase order or another source document that the customer references internally.

By default, Odoo does not include the source document field in your search. You must use the advanced search function each and every time in order to look up a customer's order by the source document they have provided you with. As in much of this book, this example comes from a real-world scenario. When customers call, they often won't have an invoice or sales order number from your company; instead, they may only have their internal source document. Let's see how we can customize Odoo to search the **Source Document** field for a sales order more efficiently.

Specifying additional fields

One of the greatest features of Odoo's developer tools is that you can hover over any field in a form and see important information about that field. For our example, we are going to hover over the **Source Document** field to learn about how this field is represented within Odoo's database as shown below:

The screenshot shows the Odoo Sales Quotations interface for a record named SO003. The top navigation bar includes links for Sales, Orders, To Invoice, Products, Reporting, Configuration, and a user menu for Bob Sacamano. The main content area displays the quotation details:

Customer	Mike Smith	Confirmation Date	01/16/2019 15:41:48
Invoice Address	Mike Smith	Pricelist	Public Pricelist (USD)
Delivery Address	Mike Smith	Payment Terms	15 Days
		Date Required	04/11/2019

Below this, there are tabs for Order Lines, Optional Products, and Other Information, with Other Information being active. The Shipping Information section shows a Shipping Policy of "Deliver each product when available". The Sales Information section lists Salesperson (Bob Sacamano), Tags, Sales Team (Sales), and Customer Reference. A tooltip for Customer Reference points to a Source Document field with the value "Opportunity". The Invoicing section shows Fiscal Position and Invoice Status (Fully Invoiced). The Marketing section lists Campaign, Medium, and Source.

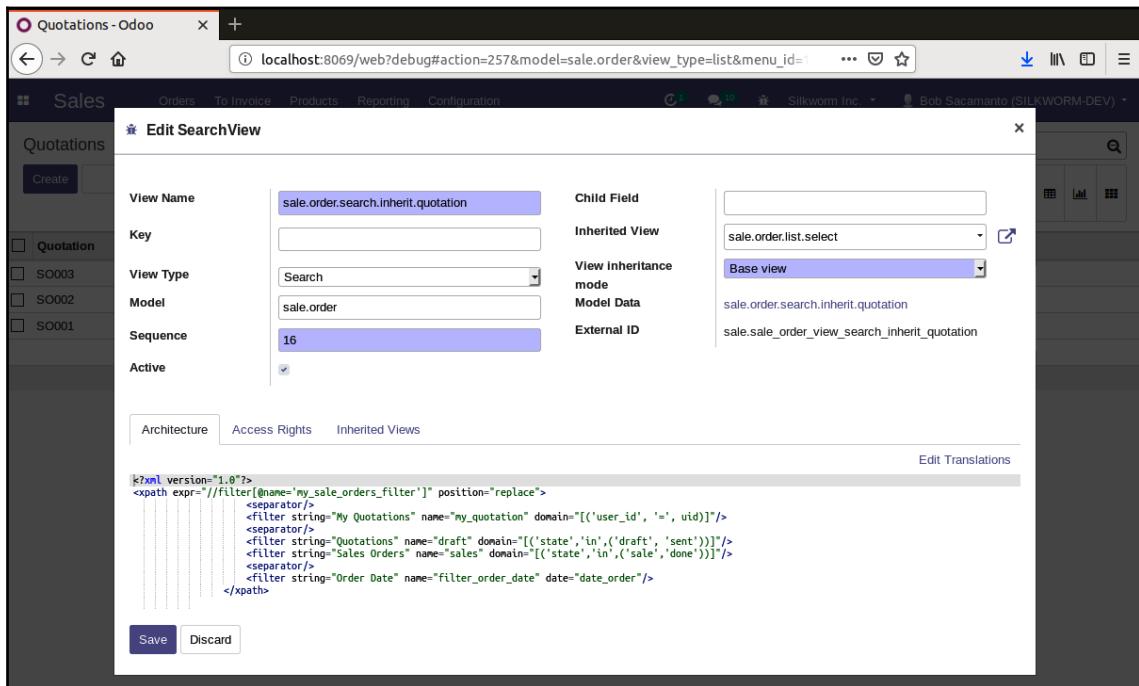
As you can see, the **Field** name for the **Source Document** in the sales order is **origin**. We can also verify that this field belongs to the **sale.order** object and it is of the **char** type.

We will use this information to modify the search view so that we can add the ability to search for a source document in the list view without using an advanced search.

Editing the search view

We can edit the search view by navigating to the list for the search we wish to modify. In this case, we will simply click on **Sales Orders** under the main **Sales** menu. Then, under the **Debug View** menu, we will choose **Edit SearchView**.

The XML for the search view will appear for us to edit, as shown in the following screenshot:



Notice that there is an inherited view specified. This means that the view you are looking at is inherited from **sale.order.list.select**. Click on the small icon to the right of the **Inherited View** field to pull up the view it is inherited from.



A view that is inherited is a view that is based on another view. With this, Odoo allows you to extend the existing views and forms with your own content. You inherit the features of the view it is based on.

Take a little time to look at the structure of this form. You will see that many tags start with `<field name="".` Each of these fields defines what search fields are available to you in the list's search box.

To search **Source Document**, we only need to add a `field` tag for the `origin` field to the list. Here, you can see that we have added it under the `team_id` field:

Open: Inherited View

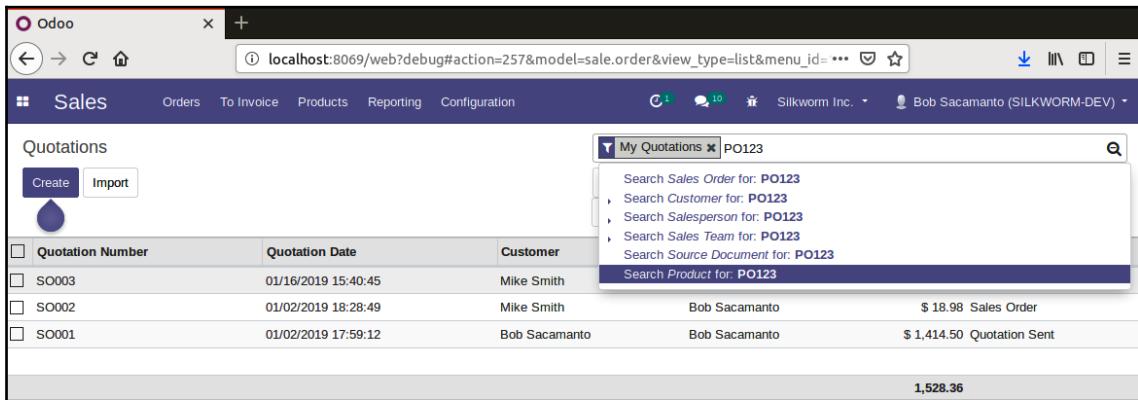
View Name	sale.order.list.select	Child Field	
Key		Inherited View	
View Type	Search	View inheritance mode	Base view
Model	sale.order	Model Data	sale.order.list.select
Sequence	15	External ID	sale.view_sales_order_filter
Active	<input checked="" type="checkbox"/>		

Architecture Access Rights Inherited Views Edit Translations

```
<?xml version="1.0"?>
<search string="Search Sales Order">
    <field name="name" string="Sales Order" filter_domain="['|','|',('name','ilike',self),('client_order_ref','ilike',self),
    <field name="partner_id" operator="child_of"/>
    <field name="user_id"/>
    <field name="team_id" string="Sales Team"/>
    <field name="origin"/>
    <field name="analytic_account_id" groups="analytic.group_analytic_accounting"/>
    <!-- We only allow to search on the following sale order line fields (product, name) because the other fields, such as p
        will not be searched as often, and if they need to be searched it's usually in the context of products
        and then they can be searched from the page listing the sale order lines related to a product (from the product itse
    ->
```

Save Discard

After you have saved the form, refresh the page. If you begin typing in the search box, you will now see that **Source Document** is available in the search list:



The screenshot shows the Odoo Sales Quotations list view. At the top, there's a search bar with the placeholder "localhost:8069/web?debug#action=257&model=sale.order&view_type=list&menu_id=***". Below the search bar is the main menu bar with Sales, Orders, To Invoice, Products, Reporting, Configuration, and a user dropdown for Bob Sacamanto (SILKWORM-DEV). The main content area displays a list of quotations with columns for Quotation Number, Quotation Date, Customer, and some additional details. On the right side of the list, there's a sidebar titled "My Quotations" with a search input field containing "PO123". A dropdown menu is open under the search field, listing several search options: "Search Sales Order for: PO123", "Search Customer for: PO123", "Search Salesperson for: PO123", "Search Sales Team for: PO123", "Search Source Document for: PO123", and "Search Product for: PO123". The "Search Product for: PO123" option is highlighted with a blue background.

As you can see, the **Source Document** is added to the search under **Sales Team**.



When making changes in these forms, all of the previous warnings apply. Make sure you do not make changes in live systems, and also make sure you have good backups.

Understanding actions

We've already seen how we can modify views and create custom search criteria, so you should be well on your way to making customizations that are quite useful. At some point, however, when you wish to create a new view or a specific type of filter, you need to understand actions so that you can change the way an Odoo application behaves.

As we learned earlier, we can use actions to trigger views. Let's begin by looking at the list of actions that are already in our Odoo installation. While in developer mode, go to **Settings** and under the technical menu, choose **Window Actions**. Now type **sales order** into the search box and hit *Enter* on your keyboard. We will be presented with a list, as shown in the following screenshot:

Action Name	Destination Model	View Type	View Ref.	Domain Value	Context Value
Create a Sales Order	project.create.sale.order	Form	project.create.sale.order.wizard.form	[]	{}
Sales Orders	sale.order	Form		[('state', 'not in', ('draft', 'sent', 'cancel'))]	{}
Sales Orders	sale.order	Form		[('state','not in',(‘draft’,‘sent’,‘cancel’))]	{‘search_default_team_id’: [active_id], ‘default_team_id’: active_id, }
Sales Orders	sale.order	Form		[('state', '=', 'sale'), ('invoice_status', '=', 'to invoice')]	{‘search_default_team_id’: [active_id], ‘default_team_id’: active_id, }

In the previous screenshot, we filtered the list of actions to only those named **Sales Orders**. Now, let's create our own custom rush order action, which will allow us to create a menu that will only pull up the sales orders that are rush orders. This is a perfect example of simple customization that can save a lot of keystrokes and improve usability, depending on the operation.

One nice thing about customizing in Odoo is that you can often use an existing record as a template and then simply make the custom changes you require. This dramatically reduces the risk of making any data entry errors.



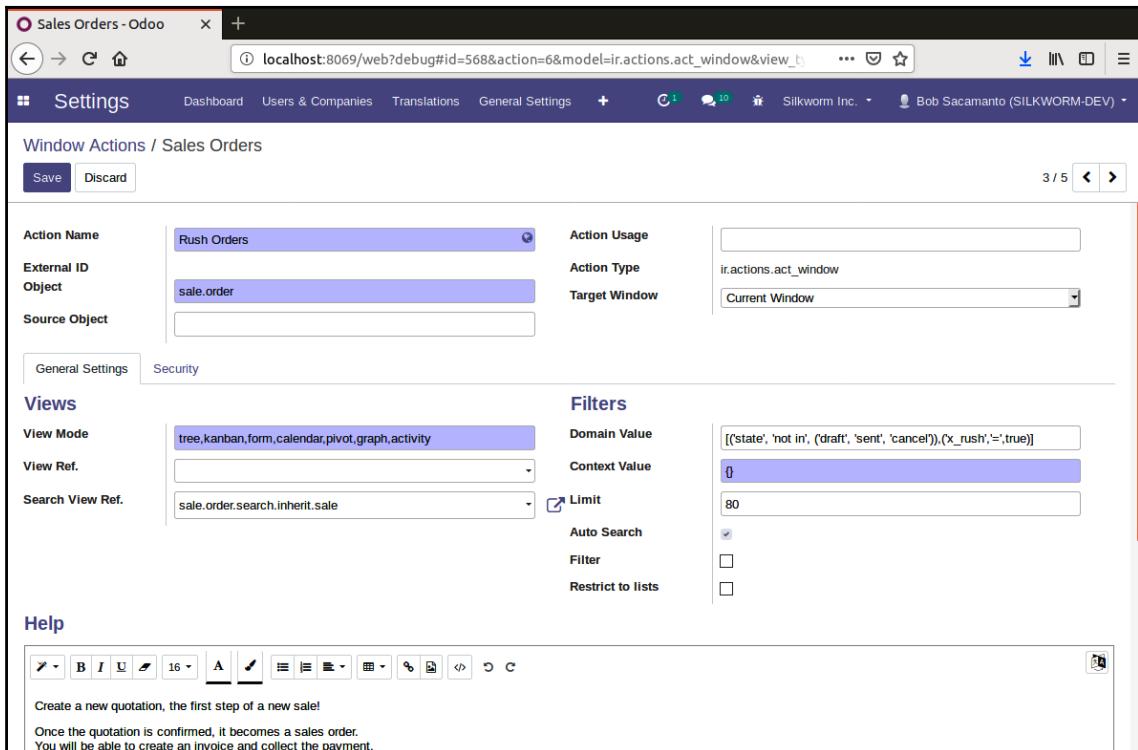
When you are first starting out, it is recommended that you consider making one change at a time and testing it out before making additional customization.

Let's duplicate the first sales orders action in the list that has the `[('state', 'not in', ('draft', 'sent', 'cancel'))]` domain value.



A domain value is the technical name for the filter you supply to limit the records.

Choose **Duplicate** from the **Action** drop-down menu. Change the **Action Name** to **Rush Orders**, as shown in the following screenshot:



Notice that after we duplicated the action, we changed the action name so that we can recognize that this action will only show orders that have been designated as rush orders.

To actually filter the `sale.order` model by orders that have been marked as rush orders, we need to change the domain value of our new action. Notice how the domain value already has a limit that says this cannot be a draft order. We will leave that filter in place and add an additional condition specifying that the order must also be a rush order.

In a large operation, a simple modification like this could reduce a screen that has thousands, or even hundreds of thousands, of records, to a much more manageable size for the specific process.

Make sure you save the changes made to your new action. Now, you can add your new menu.

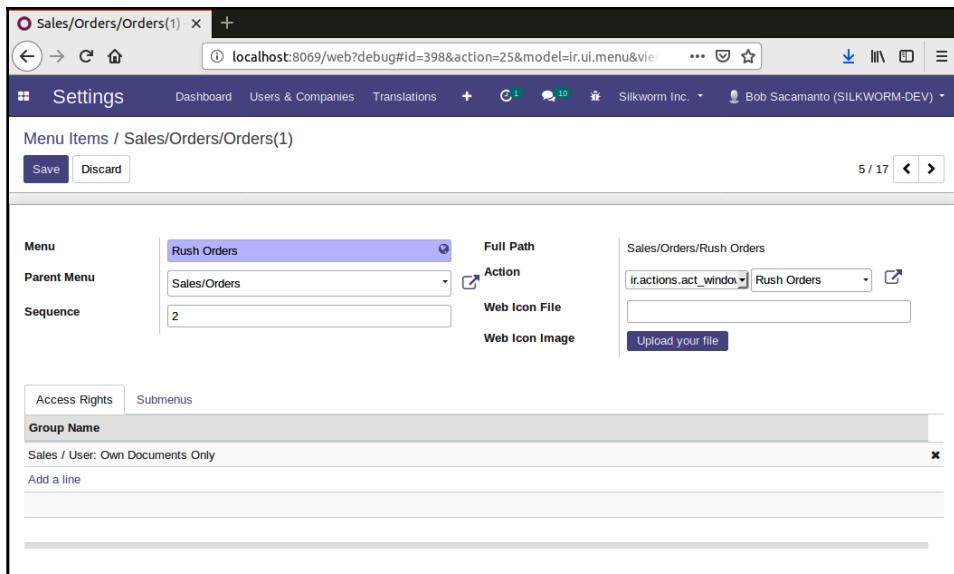
Creating a new menu

When customizing Odoo, having to create a new menu that will pull a new view or an action that will filter that view, is inevitable. This ability to create new menus and tie them to your own custom actions can create a better user experience that is more customized to your specific business requirements.

While in developer mode, go to the **Settings** menu and choose **Menu Items** under **User Interface** as shown in the following screenshot:

The screenshot shows the Odoo developer mode interface with the URL `localhost:8069/web?debug#action=25&model=ir.ui.menu&view_type=list&menu_id=4`. The top navigation bar includes 'Settings', 'Dashboard', 'Users & Companies', 'Translations', 'General Settings', and 'Technical'. On the right, there are icons for 'C 1', '10', 'Silkworm Inc.', and 'Bob Sacamano (SILKWORM-DEV)'. The main content area is titled 'Menu Items' with 'Create' and 'Import' buttons. A search bar and filters are at the top right. The list below shows a large number of menu items, all of which have a plus sign icon next to them, indicating they are expandable. The list includes categories like 'Settings/Dashboard', 'Dashboards/Dashboards', 'Settings/Users & Companies', 'Settings/Users & Companies/Users', 'Invoicing/Configuration/Analytic Accounting/Analytic Accounts', 'Invoicing/Configuration/Settings', 'Sales/Configuration/Settings', 'CRM/Configuration/Settings', 'CRM/Configuration/Pipeline/Stages', 'Purchase/Purchase/Requests for Quotation', 'Purchase/Configuration/Settings', 'Inventory/Overview', 'Inventory/Configuration/Settings', 'Manufacturing/Configuration/Settings', 'Invoicing/Reporting/France', 'Employees/Human Resources', 'Employees/Configuration/Settings', 'Project/Configuration/Settings', 'Timesheets/Configuration/Settings', 'Recruitment/Configuration/Settings', 'Settings/Translations', 'Settings/Technical/Email', and 'Settings/Technical/User Interface/Menu Items'.

Like before, we can use the search feature to limit the menu to the sales order's menu items. Just as we duplicated the sales order's action to make it easier to create a custom action, duplicate the sales order menu item as a starting point for your new **Rush Sales Order** menu, as shown in the following screenshot:



Notice how we have changed the name that will appear in the menu and have assigned the **Sales Order - Rush** action to this menu. Odoo now knows that when we select the **Sales Order - Rush** menu item, the **Sales Order - Rush** action will be triggered. Because we modified the domain value of the action to only include rush orders, we only see the records that match that criteria.

Integrating applications using Odoo's API

Sometimes, you want to connect to Odoo's data from an external application. The Odoo API is useful for these scenarios. It utilizes XML-RPC, which is a protocol for remote procedure calls, and it uses XML syntax to connect with the data on an Odoo server.

You can access Odoo's API via languages such as Python, Ruby, PHP, and Java. Let's look at an example in Python 3. (If you don't have Python installed already, go to <http://www.python.org/download>.) Pull up your Python Terminal and type the following:

```
import xmlrpclib
info = xmlrpclib.ServerProxy('https://demo.odoo.com/start').start()
```

```
url, db, username, password=\n    info['host'], info['database'], info['user'], info['password']
```

If you've been following along, you'll know that you've just created an instance of Odoo's API client, and then had that client create an instance of Odoo at the `demo.odoo.com` website. Now, let's connect, authenticate, and retrieve our user ID (`uid`) from the server:

```
common = xmlrpclib.ServerProxy('{}/xmlrpc/2/common'.format(url))\nuid = common.authenticate(db, username, password, {})\nprint(uid)
```

We'll need that `uid` to connect to the models and pull data from them. So now, let's create a connection to the `res.partner` model and call one of its methods using the `execute_kw` RPC function (the `kw` stands for **keyword**, denoting that you will pass a dictionary variable as the final parameter). This time, we will request a list of partners who are customers and companies, rather than individuals as follows:

```
model = xmlrpclib.ServerProxy('{}/xmlrpc/2/object'.format(url))\nmodel.execute_kw(db, uid, password, 'res.partner',\\\n    'search', [[[{'customer': '=', 'is_company': '=', 'True}]]])
```

The result you see will be a list of IDs for all matching `res.partner` records:

Summary

In this chapter, we started by looking at how to activate Odoo's developer mode. Next, we walked through how to make a backup of the Odoo database and how to restore that database using database management tools in Odoo, while emphasizing the importance of creating backups. Next, we went through the step-by-step process of customizing Odoo, looking at how to add fields to the database and ultimately, to your forms and views. Finally, we queried data using Odoo's API. This is by no means a comprehensive guide to Odoo customization; it is merely an introduction to common ways of customizing forms, lists, and searches.

In the next chapter, we will explore how we can use the Odoo report designer to customize reports and export data from Odoo. We will begin by looking at how to customize the company headers and footers that appear on documents. With Odoo, we can use dynamic fields to automatically put values from our database into our reports. Using the new QWeb template language, we will get a great deal of power and flexibility when building reports that integrate well with Odoo.

14

Modifying Documents and Reports

Regardless of how great the built-in reports are in any **Enterprise resource planning (ERP)** system, it is inevitable that most companies will need to do some custom modifications to the standard documents and reports that are provided. Of course, Odoo is no exception to this rule. The goal of this chapter is to provide you with a solid introduction to the Odoo reporting framework.

In this chapter, we will cover the following topics:

- Overview of report modification, QWeb templates and printing to PDFs
- Making simple changes to the headers and footers of your reports
- Basic organization of reports and forms and their various types
- Understanding and modifying structure under the existing Sales order form
- Modifying reports using the Odoo reporting framework and the QWeb template language

Within Odoo, it is possible to make some changes without modifying the documents themselves. For example, you can simultaneously change the headers and footers that appear on all your reports throughout the company.

A powerful template language called QWeb allows you to integrate data from Odoo into your reports. This chapter will walk you through these steps and show you how to modify existing Odoo reports.



As in other areas of Odoo development, be sure to make frequent backups of your database. In some of the examples in this chapter, we will change the database in ways that can make it difficult to recover if something goes wrong.

Developing the skills that are required to modify reports

In my experience, many end users believe that with perhaps an hour or two of training, they will be able to create their own reports. This is not just an Odoo issue; it is a common perception that many end users have when working with any ERP system. Unfortunately, creating or modifying reports is often not as easy as it appears and should be considered much more of a developer task, rather than an end user task. Be prepared to spend considerable time practicing and acquiring the skills that are required to make significant changes to the documents and reports that are provided to you in Odoo.

Furthermore, reporting has changed dramatically in the past few years. Previously, it wasn't uncommon for many companies to be totally dependent upon paper reports and Excel sheets to properly communicate information throughout the company. Often, in Odoo, what was previously a physical report can be better managed through filtering and grouping views properly or using the business intelligence features built into Odoo. Be sure to carefully consider your report options from many different angles before you jump into modifying existing reports and documents.



Scope creep refers to projects that continue to grow in size and complexity far beyond the original design. Pay special attention to report requests as this can be an area in which you can end up with many additional requirements that you did not anticipate during the planning stages.

What is the QWeb template engine?

The **QWeb** template engine can actually be used for many things other than basic reports. Besides reports, the QWeb template is also the main way that all of the website builders and new CMSes generate the HTML to create pages. In reports, QWeb works in exactly the same way since it generates HTML; however, it directs the output into a PDF file instead of the browser page.

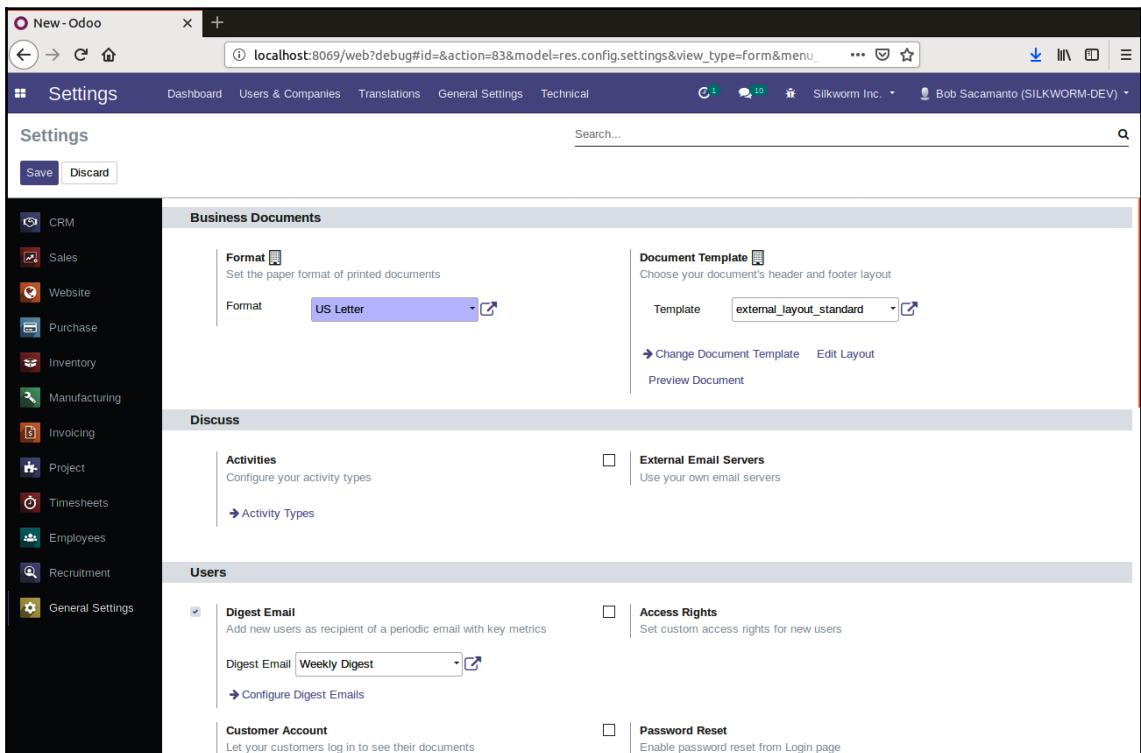
The great part about this is that once you really learn how to modify reports in QWeb; that same skill will allow you to create dynamic web pages that can tie directly into Odoo. This is a powerful reason to learn QWeb.

Company report configuration

When Odoo is first installed, you are presented with a default template that will appear on many of the standard reports. Even if you don't plan to make a lot of major changes to the standard reports in Odoo, it is very likely that you will want to modify the headers and footers, as well as other parts of your report template, so that they're more unique to your company.

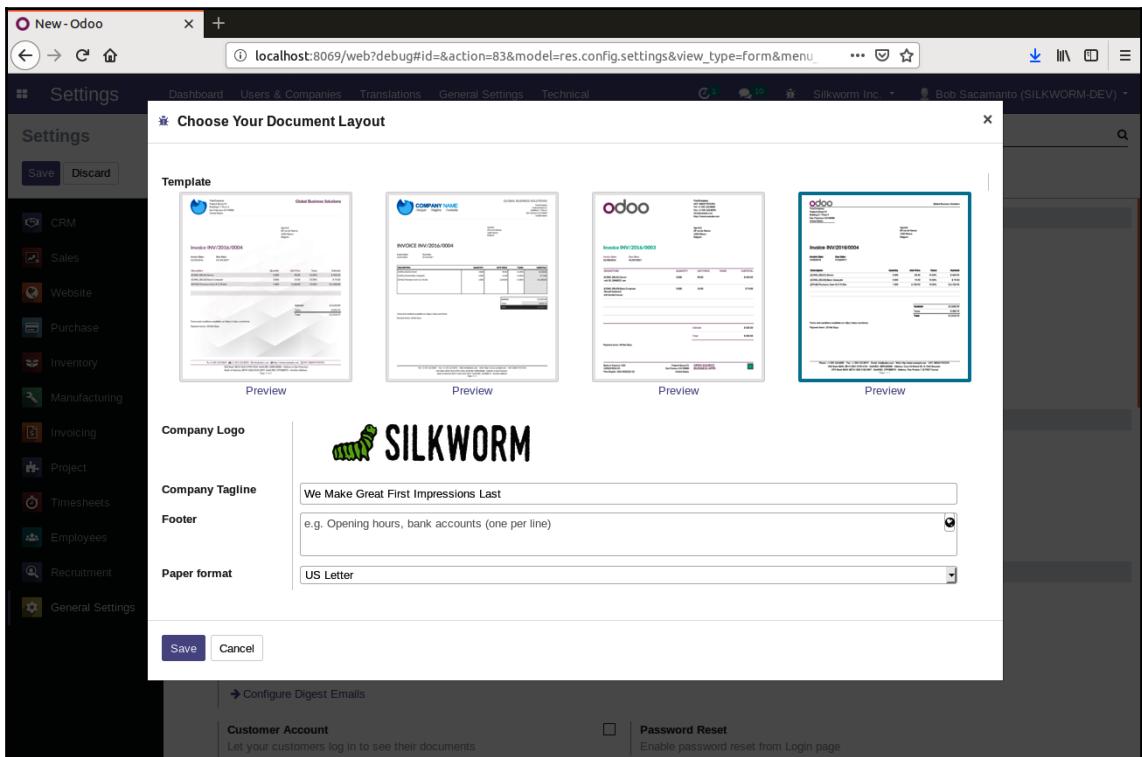
To begin editing Odoo headers and footers, it is best if you log in as an Odoo administrator. Furthermore, to access all the report settings and configurations, you will need to go into developer mode, as described in Chapter 13, *Customizing Odoo for Your Business*. This is accomplished by going to **Settings** and then clicking on the **Activate Developer Mode** link at the bottom-right of the form.

Once you have developer mode turned on, navigate to **Settings** again and click on the **General Settings** menu. At the top of the form, you will find the **Business Documents** area as shown in the following screenshot:



Before we make any changes to our reports, let's go ahead and get a look at one of the default reports.

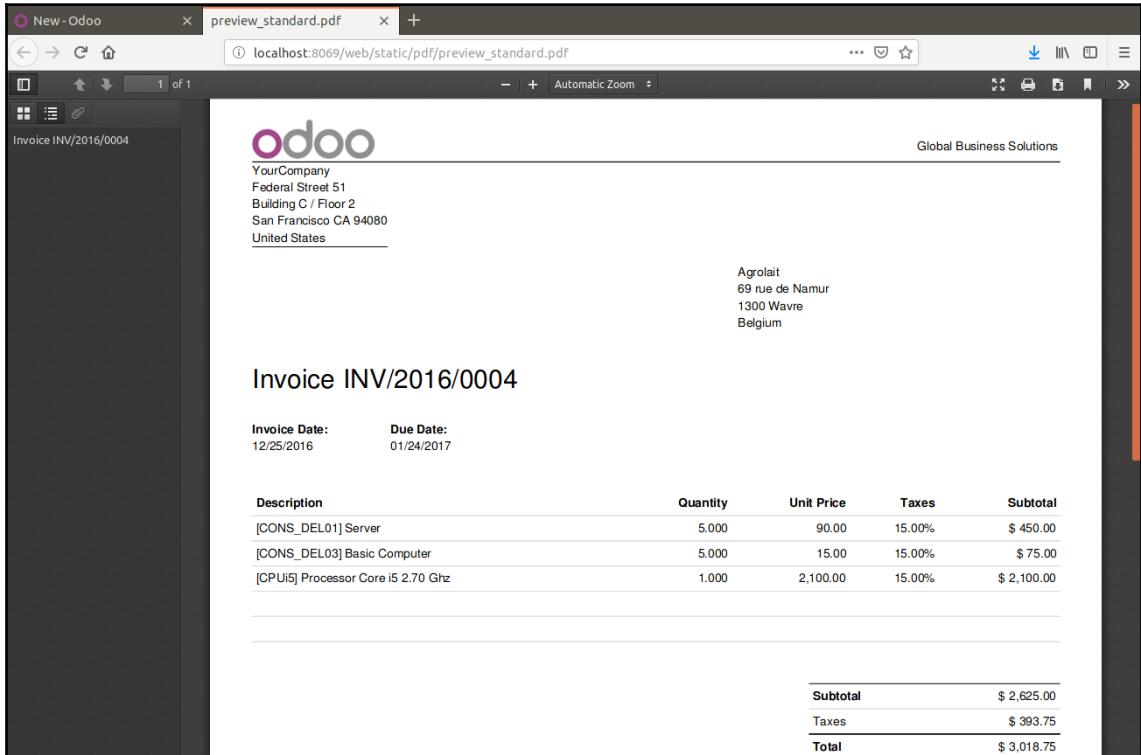
If you click on **Change Document Template**, you will be taken to a form that allows you to choose the default template for your reports. You can see what each one of them looks like by clicking **Preview** under the document templates. As shown in the following screenshot, there are four templates that you can choose from. Hovering over each **Preview** link will reveal its name in the URL; they are (from left to right) Background, Boxed, Clean, and Standard. The currently active layout, that is, standard, is highlighted with a blue rectangular outline:



These previews will not display with your logo, tagline, address, or slogan, but they will still give you a good idea of the different document layout styles that are available. They show you where on the page the company name, address, email, website, logo, tagline, and more will appear.

Odoo will then process the report into a PDF and allow you to download it in your browser for viewing.

After the PDF has been downloaded, it will pop open in a new browser tab, and you can view it to see a preview of what your reports will look like. Note that these previews don't use data from your own database, but uses the demonstration data instead:



To close the PDF preview, simply close the browser tab.

Now, you should be back on the **Choose Your Document Layout** screen. To switch to an alternative layout, simply click on one of the other document layout styles shown and then click the **Save** button to have your reports utilize that new layout style. For now, let's choose the Standard layout, which is the one on the far right.

Problems with the Wkhtmltopdf installation

While the majority of Odoo installations go smoothly for primary operations, it is quite common for installations to have problems with a library called **Wkhtmltopdf**. If you have this problem with your own installation, you will see a message like the following:



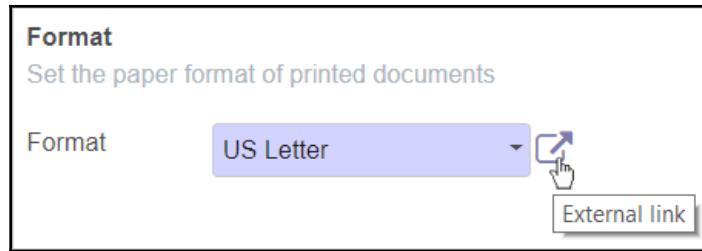
Follow the link to <http://www.wkhtmltopdf.org> and look at the resources in this book's *Appendix A, Locating Additional Odoo Resources*, to learn more about how to install this package so that Odoo can properly display the PDF files you will need for reporting purposes. Usually, fixing the problem is as simple as uninstalling the bad installation of the Wkhtmltopdf library and then reinstalling a fresh one.



Wkhtmltopdf stands for **WebKit HTML to PDF** and is comprised of a set of command-line tools that are useful for rendering HTML output into a standard PDF format using the Qt WebKit rendering engine, yet another open source project. This library of commands, if installed correctly, should run seamlessly behind the scenes. However, Odoo's installer sometimes struggles to install this package properly.

Specifying the paper format for your reports

Depending on your specific business requirements, it is possible that you will need to have custom paper formats. Fortunately, Odoo allows you to specify a default paper format, as well as indicate which reports should use a given paper format. While you may not need to use this feature, it's worth knowing how to do so, in case the need arises at a later time. To configure an alternative paper format, simply click the **External link** icon to the right of the **Format** drop-down list:



Now, you will see the options that are available to set a custom paper format:

A screenshot of the Odoo 'Create: Paper format' dialog box. It contains the following fields:

Name	Pricelist Report
Paper size	A7 11 74 x 105 mm
Orientation	Portrait
Top Margin (mm)	25.00
Bottom Margin (mm)	25.00
Left Margin (mm)	6.00
Right Margin (mm)	6.00
Display a header line	<input type="checkbox"/>
Header spacing	35
Output DPI	90
Associated reports	Pricelist <input type="button" value="X"/>

At the bottom are two buttons: 'Save' and 'Discard'.

In the preceding example, we can see a custom paper configuration that's been created for an A7-sized sheet of paper. Most importantly, note how the **Associated reports** include the **Pricelist**. This is how Odoo can associate a given report (or reports) with a specific paper size and setup.

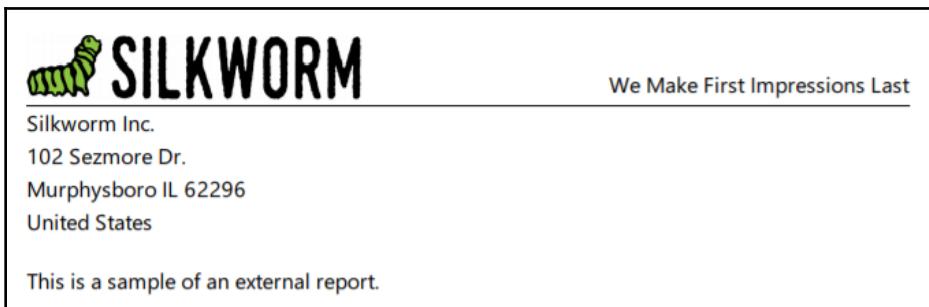


Troubleshooting printers and paper sizes can be tricky. Even if you have your settings correct in this Odoo configuration, you will need to consider the settings in the application you use to print the report (this will likely be your PDF reader), as well as the settings on the printer itself.

Making our first simple change

When modifying reports, the best approach is to start simple and test each and every change as you make it. Don't expect to go in and make a dozen changes to the header or the footer and then run the report without any nasty surprises – as least not until you have a tremendous amount of experience under your belt.

First, let's preview a report that contains our own company logo and address. Under the **Change Document Template** link, click the link that says **Preview Document**. This will download a PDF file that should look similar to the following in the header:



The sample's report footer should look similar to this:



Certain businesses may wish to include social media links on documents. With that in mind, let's dive into a bit of QWeb code to append the company's Facebook link to the footer of our reports.

First, let's make sure that we have filled in the Facebook information for our company by going to **Settings | Dashboard** and clicking on the **Set Up** button (under **Set up your company information**). Enter the Facebook link under **Social Media** if it isn't already there:

The screenshot shows the 'General Information' tab selected in the Odoo settings. The company details are as follows:

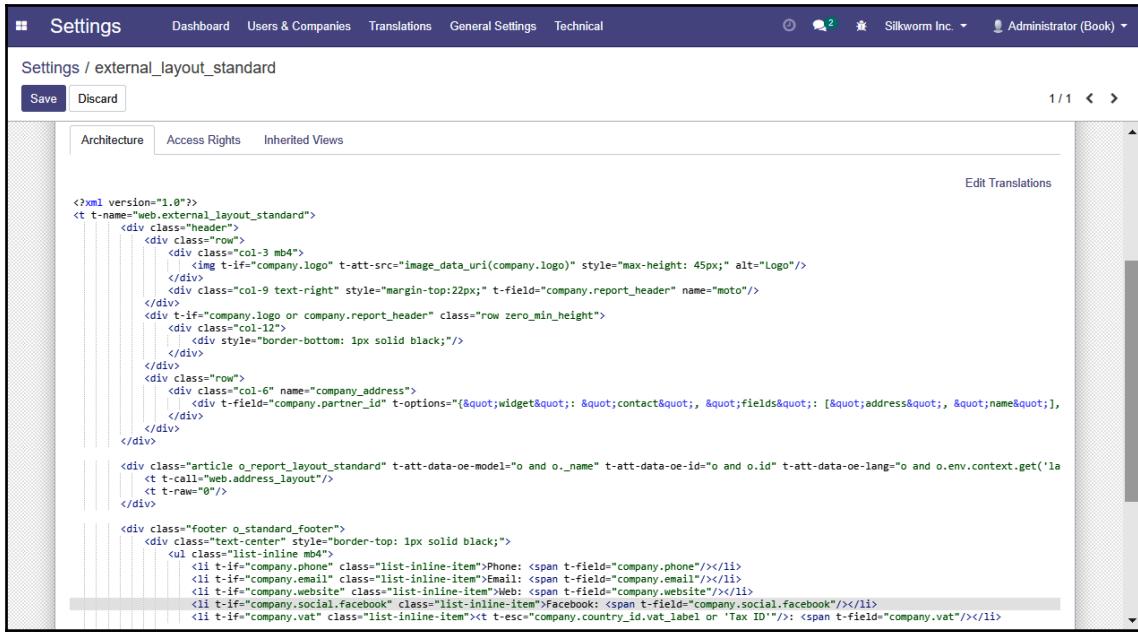
Partner	Silkworm Inc.	Website	http://silkwormink.com
Address	102 Sezmore Dr. Murphysboro Illinois (US) 62296 United States	Phone Email Tax ID Company Registry Default incoterm Currency	USD
		Nomenclature	Default Nomenclature

In the **Social Media** section, the Facebook account is listed with the URL <https://www.facebook.com/silkwormink>.

Next, access the QWeb code by going into **Settings | General Settings** with Developer Mode activated and, under **Document Template**, click on the link that reads **Edit Layout**. The actual code we will add is very simple. However, we can make it easier on ourselves by copying and pasting it, and then just changing it. If you know HTML, you can use HTML in QWeb code.

The XML code uses an `` tag to display the information based on the list-inline style. We can use the `t-if` condition to only show this markup if there is a Facebook URL in our company data. We also have a `<class>` tag, which is used to format the appearance of the link. Finally, we use a `` tag to display the actual URL from the company model.

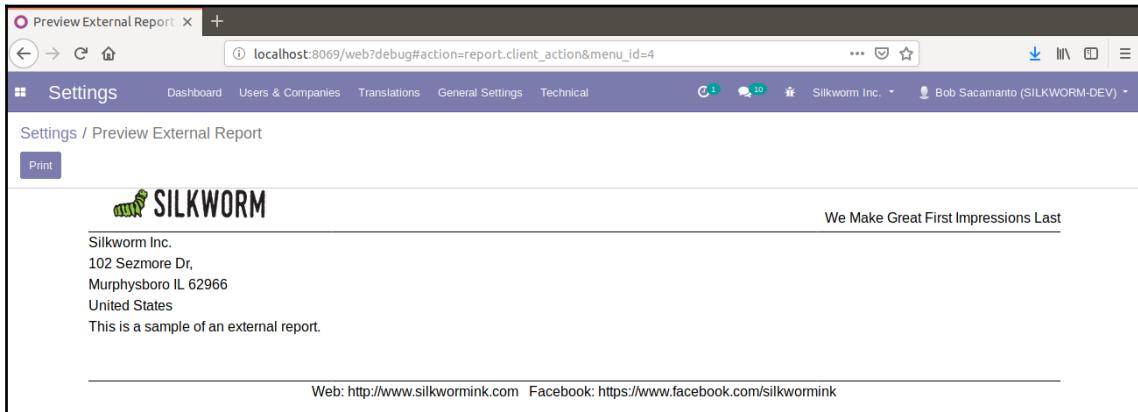
When you are done, the edited code should look as follows:



The screenshot shows the Odoo Settings module with the "external_layout_standard" configuration. The XML code is displayed in a code editor window. The code defines a layout structure with a header section containing a logo and a report header. It also includes sections for company address and footer information. A "Save" button is visible at the top left, and a "Discard" button is at the top right. The status bar at the bottom indicates "1 / 1".

```
<?xml version="1.0"?>
<t t-name="web_external_layout_standard">
    <div class="header">
        <div class="row">
            <div class="col-3 mb4">
                
            </div>
            <div class="col-9 text-right" style="margin-top: 22px;" t-field="company.report_header" name="moto">
                </div>
        </div>
        <div t-if="company.logo or company.report_header" class="row_zero_min_height">
            <div class="col-12">
                <div style="border-bottom: 1px solid black;">
                    </div>
            </div>
        </div>
        <div class="row">
            <div class="col-6" name="company_address">
                <div t-field="company.partner_id" t-options="{'widget': 'contact', 'fields': ['address', 'name']}>
                </div>
            </div>
        </div>
    </div>
    <div class="article o_report_layout_standard" t-att-data-oe-model="o" and o._name" t-att-data-oe-id="o.id" t-att-data-oe-lang="o" and o.env.context.get('lang') or 'en_US'" t-raw="0">
        <t>
            <div class="text-center" style="border-top: 1px solid black;">
                <ul class="list-inline mb4">
                    <li t-if="company.phone" class="list-inline-item">Phone: <span t-field="company.phone"/></li>
                    <li t-if="company.email" class="list-inline-item">Email: <span t-field="company_email"/></li>
                    <li t-if="company.website" class="list-inline-item">Web: <span t-field="company.website"/></li>
                    <li t-if="company.social.facebook" class="list-inline-item">Facebook: <span t-field="company.social.facebook"/></li>
                    <li t-if="company.vat" class="list-inline-item"><t-esc="company.country_id.vat_label or 'Tax ID'"/>: <span t-field="company.vat"/></li>
                </ul>
            </div>
        </t>
    </div>
</div>
```

Now, you can preview the report to see the changes:



The screenshot shows the "Preview External Report" interface. The browser address bar displays "localhost:8069/web?debug#action=report.client_action&menu_id=4". The page title is "Settings / Preview External Report". The content area shows a sample report with a header featuring the "SILKWORM" logo and the tagline "We Make Great First Impressions Last". The report lists the company's address: "Silkworm Inc., 102 Sezmore Dr, Murphysboro IL 62966 United States". Below this, it states "This is a sample of an external report". At the bottom, there are links for "Web: http://www.silkwormink.com" and "Facebook: https://www.facebook.com/silkwormink".

Remember to start with simple changes and examine other reports to learn how to make more complex changes.

Modifying the internal report header using QWeb

From the options we have seen so far, it is clear that while we can create a custom footer, we must do something else in order to change the header for our reports. To make this change, we must edit the actual QWeb source that makes up the header of the report.

Let's look at the QWeb for the company header by returning to the **General Settings** and clicking the **Edit Layout** link. The following is a screenshot of the QWeb that was used for the external report header:

View Name	external_layout_standard	Child Field	
Key	web.external_layout_standard	Inherited View	
View Type	QWeb	View inheritance mode	Base view
Model		Model Data	external_layout_standard
Sequence	16	External ID	web.external_layout_standard
Active	<input checked="" type="checkbox"/>		

Architecture Access Rights Inherited Views

Edit Translations

```
<?xml version="1.0"?>
<t t-name="web.external_layout_standard">
    <div class="header">
        <div class="row">
            <div class="col-3 mb4">
                
            </div>
            <div class="col-9 text-right" style="margin-top:22px;" t-field="company.report_header" name="moto">
                </div>
            <t-if="company.logo or company.report_header" class="row zero_min_height">
                <div class="col-12">
                    <div style="border-bottom: 1px solid black;">
                        </div>
                </div>
                <div class="row">
                    <div class="col-6" name="company_address">
                        <div t-field="company.partner_id" t-options="{'widget': 'contact', 'fields': ['address', 'name']}>
                            </div>
                    </div>
                </div>
            </t-if>
            <div class="article o_report_layout_standard" t-att-data-oe-model="o and o._name" t-att-data-oe-id="o and o.id" t-att-data-oe-lang="o and o.env.context.get('lang') or 'en'" t-call="web.address_layout"/>
            <t t-raw="0"/>
        </div>
    </div>
    <div class="footer o_standard_footer">
        <div class="text-center" style="border-top: 1px solid black;">
            <ul class="list-inline mb4">
                <li t-if="company.phone" class="list-inline-item">Phone: <span t-field="company.phone"/></li>
                <li t-if="company.email" class="list-inline-item">Email: <span t-field="company.email"/></li>
            </ul>
        </div>
    </div>
</t>
```

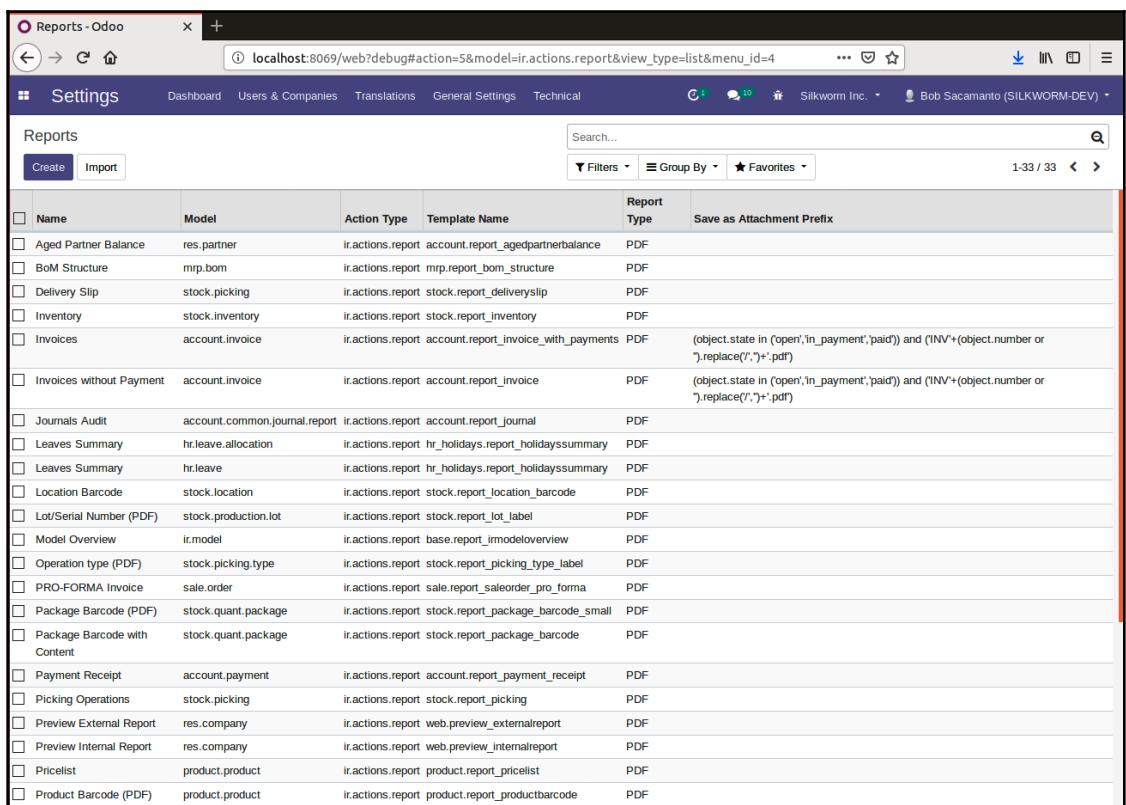
Here, we can see the actual QWeb code that displays the report header in HTML, which is then converted into a PDF report. As we mentioned earlier in this chapter, modifying reports is not something that should typically be attempted by end users. This example has been used to demonstrate how you can make a small change to a report. By analyzing other reports in Odoo, it is possible to make your own custom reports. Just be prepared for a considerable learning curve if you are new to XML and template languages.

Learning how Odoo organizes reports

Unfortunately, the ability to edit the header and footer of the company information doesn't get you very far. If you spend time with Odoo, it is inevitable that the time will come when you need to make changes to specific Odoo documents. For example, a company may need to customize their quotation or sales order to make it more visually attractive to their customers. Perhaps a company would like to change the appearance of their invoice or the picking ticket they use to pull products from the inventory.

However, before you can start modifying reports or adding new reports, it is important that you have an overall understanding of how reports are organized within Odoo. First, make sure that you have activated developer mode, as you learned to do in the previous chapter.

With developer mode active, you can get access to the reports within Odoo by going to the **Settings** menu. Then, in the **Technical** section further down in the menu, you will find the **Reports** option. Clicking this will list the **Reports** in Odoo, as shown in the following screenshot:



The screenshot shows the Odoo Reports list page. At the top, there is a header bar with the Odoo logo, a search bar containing 'localhost:8069/web?debug#action=5&model=ir.actions.report&view_type=list&menu_id=4', and various browser controls. Below the header is a navigation bar with 'Settings' selected, followed by 'Dashboard', 'Users & Companies', 'Translations', 'General Settings', and 'Technical'. On the right side of the header, there are icons for a user profile ('Silkworm Inc.'), developer mode ('Bob Sacamano (SILKWORM-DEV)'), and a message count ('1').

The main content area is titled 'Reports' and contains a table with the following columns: Name, Model, Action Type, Template Name, Report Type, and Save as Attachment Prefix. The table lists 25 different reports, each with a checkbox next to it. The 'Report Type' column for most reports is 'PDF'. Some reports have complex template names and specific conditions listed in the 'Save as Attachment Prefix' column.

Name	Model	Action Type	Template Name	Report Type	Save as Attachment Prefix
Aged Partner Balance	res.partner	ir.actions.report	account.report_agedpartnerbalance	PDF	
BOM Structure	mrp.bom	ir.actions.report	mrp.report_bom_structure	PDF	
Delivery Slip	stock.picking	ir.actions.report	stock.report_deliveryslip	PDF	
Inventory	stock.inventory	ir.actions.report	stock.report_inventory	PDF	
Invoices	account.invoice	ir.actions.report	account.report_invoice_with_payments	PDF	(object.state in ('open','in_payment','paid')) and ('INV'+(object.number or '').replace('.','.')+'.pdf')
Invoices without Payment	account.invoice	ir.actions.report	account.report_invoice	PDF	(object.state in ('open','in_payment','paid')) and ('INV'+(object.number or '').replace('.','.')+'.pdf')
Journals Audit	account.common.journal.report	ir.actions.report	account.report_journal	PDF	
Leaves Summary	hr.leave.allocation	ir.actions.report	hr.holidays.report_holidayssummary	PDF	
Leaves Summary	hr.leave	ir.actions.report	hr.holidays.report_holidayssummary	PDF	
Location Barcode	stock.location	ir.actions.report	stock.report_location_barcode	PDF	
Lot/Serial Number (PDF)	stock.production.lot	ir.actions.report	stock.report_lot_label	PDF	
Model Overview	ir.model	ir.actions.report	base.report_irmodeloverview	PDF	
Operation type (PDF)	stock.picking.type	ir.actions.report	stock.report_picking_type_label	PDF	
PRO-FORMA Invoice	sale.order	ir.actions.report	sale.report_saleorder_pro_forma	PDF	
Package Barcode (PDF)	stock.quant.package	ir.actions.report	stock.report_package_barcode_small	PDF	
Package Barcode with Content	stock.quant.package	ir.actions.report	stock.report_package_barcode	PDF	
Payment Receipt	account.payment	ir.actions.report	account.report_payment_receipt	PDF	
Picking Operations	stock.picking	ir.actions.report	stock.report_picking	PDF	
Preview External Report	res.company	ir.actions.report	web.preview_externalreport	PDF	
Preview Internal Report	res.company	ir.actions.report	web.preview_internalreport	PDF	
Pricelist	product.product	ir.actions.report	product.report_pricelist	PDF	
Product Barcode (PDF)	product.product	ir.actions.report	product.report_productbarcode	PDF	

Here, you can see the list of reports in the view, along with critical information that tells you which model the report is associated with, the type of action that's used to trigger that report, the template for the report, and the report type.



The **Save As Attachment** prefix can be used to append a prefix to the beginning of the report name when it's saved.

Understanding the report types

Each of the past few major upgrades to Odoo has brought new improvements to the reporting engine. Odoo 12 is no different, and many of the old reports are gone. However, Odoo still provides support for those older types of reports. In the previous screenshot, when you see a report type labelled as RML or PDF (deprecated), this is a report that is still using the old reporting mechanism. You can still open up those older reports and make changes to them. However, most of the reports now use Odoo's new Odoo reporting framework, QWeb templates.

Looking at the definition for the Sales order form

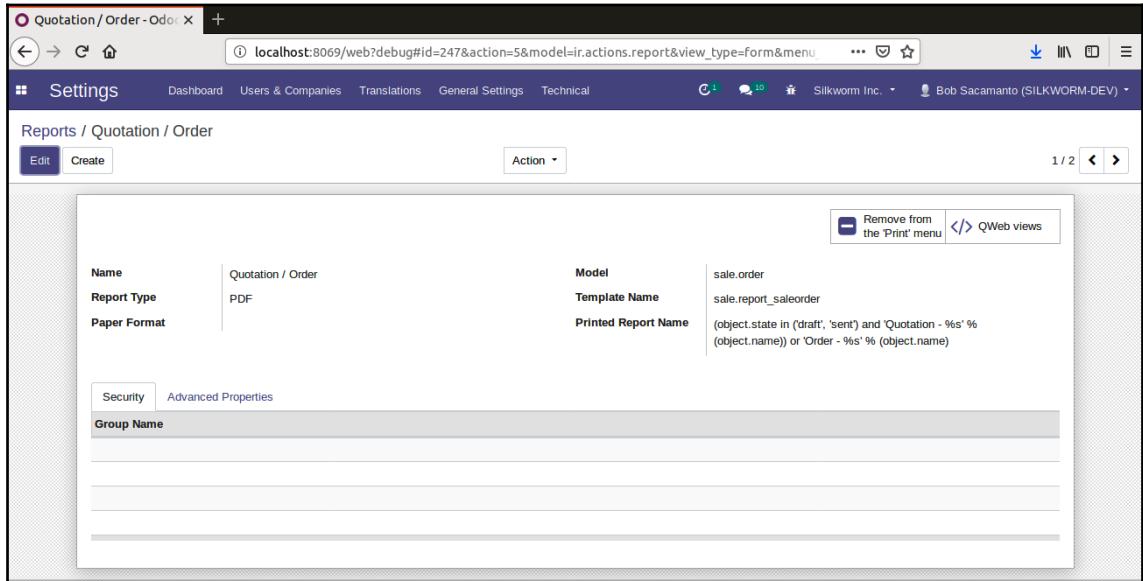
Like the other examples in this book, we are going to choose a very common business requirement for this example. Most companies are not going to want to use the default sales order. In fact, after working with hundreds of accounting and ERP systems, I cannot recall any system implementation that involves sales orders where the sales order wasn't customized at some point.

Using what you have learned from the previous chapters, you may wish to bring up a quotation or a sales order and have it ready to print so that you can see your changes as they happen. Any changes you make will modify both the quotation and the sales order, since they share the same QWeb template.



Odoo is quite good at allowing you to use more than one tab in your browser for most operations. I often keep one tab open with the document I want to print and then keep another tab open with the report I am editing.

When you are ready to edit the sales order, scroll down the report list, find **Quotation | Order**, and click on it. It is also the only report in the default list of Odoo reports that is built on the **sale.order** model as shown in the following screenshot:



You will notice that there is some basic information at the top of the page and a **Security** tab at the bottom of the report that is empty. You can use this tab to put additional restrictions on the report, beyond what has already been specified in the **sale.order** model.

The previous screen is primarily the configuration of the report. The views that make up the actual report itself can be found by clicking **QWeb views**:

The screenshot shows the Odoo web interface with the title 'Views - Odoo'. The top navigation bar is identical to the previous screenshot. Below the title, it says 'Reports / Quotation / Order / Views' with 'Create' and 'Import' buttons. The main content area is a table listing the QWeb views:

	View Name	View Type	Model	External ID	Inherited View
<input type="checkbox"/>	report_saleorder	QWeb		sale.report_saleorder	
<input type="checkbox"/>	report_saleorder_document	QWeb		sale.report_saleorder_document	
<input type="checkbox"/>	report_saleorder_document_inherit_sale_management	QWeb		sale_management.report_saleorder_document_inherit_sale_management	report_saleorder_document
<input type="checkbox"/>	report_saleorder_document_inherit_sale_stock	QWeb		sale_stock.report_saleorder_document_inherit_sale_stock	report_saleorder_document
<input type="checkbox"/>	report_saleorder_pro_forma	QWeb		sale.report_saleorder_pro_forma	

This brings up all the **QWeb views** associated with the report. While there are three views listed, the one we are interested in is **report_saleorder_document**. The **report_saleorder** view is basically a container that holds the content of the document view, and you will rarely need to modify this unless you are an experienced Odoo developer.

Click on the **report_saleorder_document** to open the view and you will get to the following screenshot:

View Name	report_saleorder_document	Child Field	
Key	sale.report_saleorder_document	Inherited View	
View Type	QWeb	View inheritance mode	Base view
Model		Model Data	report_saleorder_document
Sequence	16	External ID	sale.report_saleorder_document
Active	<input checked="" type="checkbox"/>		

Architecture Access Rights Inherited Views

```
<?xml version="1.0"?>
<t t-name="sale.report_saleorder_document">
  <t t-call="web.external_layout">
    <t t-set="doc" t-value="doc.with_context({'lang':doc.partner_id.lang})"/>
    <t t-set="address">
      <div t-field="doc.partner_id" t-options="{'widget': 'contact', 'fields': ['address', 'name'], 'no_mailing_label': true}>
        <p><t-if="doc.partner_id.vat"><t-esc="doc.company_id.country_id.vat_label or 'Tax ID'" /></t-if> <span t-field="doc.partner_id.vat"/></p>
      </t-set>
    <t-if="doc.partner_shipping_id == doc.partner_invoice_id">
      <t-set="information_block">
        <strong><t-if="doc.partner_shipping_id == doc.partner_invoice_id">Invoicing and shipping address:</t-if></strong>
        <strong><t-if="doc.partner_shipping_id != doc.partner_invoice_id">Invoicing address:</t-if></strong>
        <div t-field="doc.partner_invoice_id" t-options="{'widget': 'contact', 'fields': ['address', 'name']}>
          <t-if="doc.partner_shipping_id == doc.partner_invoice_id">
            <strong>Shipping address:</strong>
            <div t-field="doc.partner_shipping_id" t-options="{'widget': 'contact', 'fields': ['address', 'name']}>
              <t-if="not (env.context.get('proforma', False) or is_pro_forma)">
                <span t-if="doc.state not in ['draft', 'sent']">Order # </span>
                <span t-if="doc.state in ['draft', 'sent']">Quotation # </span>
              </t-if>
              <t-if="env.context.get('proforma', False) or is_pro_forma">
                <span>Pro-Forma Invoice # </span>
              </t-if>
            <span t-field="doc.name"/>
          </t-if>
        </div>
      </t-set>
    </t-if>
  </t-call>
</t>
```

We are finally here! You have to drill down a little bit to get to the report template code, but after you have done this a few times, it really is pretty easy. Now, we are looking at the actual QWeb report template. Odoo calls these templates "views" when they are associated with Odoo reports.

If you look through the architecture, you should quickly find elements within it that compare directly to the standard quotation or sales order report.

Here is a sample quotation that was produced using the default QWeb template we are viewing:

The screenshot shows a web browser displaying an Odoo application. The title bar says "Quotation / Order - Odoo". The address bar shows "localhost:8069/web?debug#action=report.client_action&menu_id=138". The top navigation bar has tabs for Sales, Orders, To Invoice, Products, Reporting, and Configuration. On the right, it shows "Silkworm Inc." and "Bob Sacamanto (SILKWORM-DEV)". Below the header, it says "Quotations / SO003 / Quotation / Order". There is a "Print" button. The main content area has the "SILKWORM" logo and the slogan "We Make Great First Impressions Last". It shows the company information for Silkworm Inc.: 102 Sezmore Dr, Murphysboro IL 62966, United States. It also shows the contact information for Mike Smith: 444 South Main, Marion IL 62959. The order number is "Order # SO003". Below that, it lists the date ordered (01/16/2019 23:41:48), salesperson (Bob Sacamanto), and payment terms (15 Days). A table details the order items:

Description	Quantity	Unit Price	Taxes	Amount
Medium White T-Shirt	5.000 Unit(s)	16.50	15.00%	\$ 82.50
			Subtotal	\$ 82.50
			Tax 15%	\$ 12.38
			Total	\$ 94.88

At the bottom, it says "Payment terms: 15 Days".

So, as we should always do, let's make a small change and see the result. I cannot emphasize enough how important it is to back up frequently and make small changes when you are first getting started.

Let's assume that for this change we want to have `Sales Order #` instead of `Order #` on our report.

This is relatively simple. Click **Edit**, as you would on any Odoo form, to edit the template. Then, scroll down the template until you find `Quotation #`:

```
<t t-if="not (env.context.get('proforma', False) or is_pro_forma)">
    <span t-if="doc.state not in ['draft','sent']">Order # </span>
    <span t-if="doc.state in ['draft','sent']">Quotation # </span>
</t>
```

Replace Order # with Sales Order #, and then save the document. Be very careful of the changes you make until you understand XML. A less common but worst-case scenario is that the mistake you make will still allow the document to save, but it will be broken.



Because the XML is represented with ordinary text, a little trick you can use is to copy and paste the XML into a text editor. By doing this, you will have a quick backup in case you make a mistake or don't get the results you expect.

After saving, you should be able to print the document and see your changes when you print the order again:

Sales Order # SO003

Description	Quantity	Unit Price	Taxes	Amount
Medium White T-Shirt	5.000 Unit(s)	16.50	15.00%	\$ 82.50
Subtotal				\$ 82.50
Tax 15%				\$ 12.38
Total				\$ 94.88

Payment terms: 15 Days

We have made that small change successfully and changed it, but, even just knowing how to do this will often allow you to add many things you need to a report.

Creating a new QWeb report in Odoo

As we learned in Chapter 13, *Customizing Odoo for Your Business*, you are almost never better off creating a blank record, and instead are better served by duplicating an existing record that is close to what you want. For example, we duplicated the Sales Order action and menu to create our custom Sales Order – Rush options.

Let's do the same thing for a new custom Odoo report that you can then tie to an action using the skills you learned in Chapter 13, *Customizing Odoo for Your Business*.

Simply go to the **Settings** menu and pull up the **Report** option to list the available reports. Use the search filter to locate the **Sales Order** form and duplicate it.

Learning more about the power of QWeb templates

Now that we have learned to make that small change, let's look a little closer at how Odoo is able to use the exact same template for both the quotation and the sales order. In fact, you may have already figured out how Odoo does this from looking at the code we modified a little bit.

One of the best things you can do to learn how to modify reports is to look at existing reports within Odoo and see how they accomplish what you wish to accomplish. When you're trying to solve a problem, see if you can find another report in Odoo that is already doing something similar to what you want.



It is often a lot easier to copy and paste part of a template you need and then change it rather than trying to get all the < . /?> syntax straight and just right. It's easier if you are working from a reference point.

Using a t-if to create a dynamic QWeb template report

Much of the power of QWeb allows you to conditionally show information, depending on various fields and information within the document. In this case, Odoo is looking at the status of the order to determine whether the template should have a label for Quote or Sales Order. Let's take a close look at the syntax so that we can understand exactly what it means.

If you don't understand at least some basic XML, now would be a good time to look at some of the available resources we have listed in the [Appendix A, Locating Additional Odoo Resources](#), or do a simple Google search for an XML tutorial. The following is an XML code snippet:

```
<span t-if="o.state not in ['draft','sent']">Sales Order # </span>
<span t-if="o.state in ['draft','sent']">Quotation #:</span>
```

You will notice that each section is wrapped in its own set of `` tags. Then, right after the first `span` tag starts, we have the `t-if=` condition. Everything between the double quotes is the condition that will determine whether what is included between the `span` tags will print.

In this specific case, `o` represents the order. We use the period, or dot notation to specify which field we want to check. In this case, it is the `state` field. Even if you are not a programmer, if you read it out and ignore the confusing syntax and punctuation, it should start to make sense. The only difference between each of these `t-if` statements is the word `not`, which follows `o.state`. When the state of the order is not in draft format or sent, then it is an order. When the state of the order is in draft format or sent, then we have a quote.

If XML and programming is very new to you, some of this may be a bit confusing. However, if you take some time to look at existing reports and use some of the resources in the [Appendix A, Locating Additional Odoo Resources](#), to learn more, you will be customizing Odoo reports in no time.

Summary

In this chapter, we started by walking through how to change the templates on company reports through Odoo in order to modify the header of our report. Next, we learned about Odoo's reporting system and how reports are organized within Odoo. Finally, we learned how to use the powerful QWeb template language to make changes to reports.

In the next chapter, we will explore how to build our own custom applications in Odoo.

15

Discovering Custom Odoo Modules

While Odoo has a lot of built-in and community modules, it is inevitable that there will be quite a few businesses that will have requirements that will be difficult to meet with the modules that are currently available. The Odoo framework offers developers the ability to extend Odoo to accomplish business objectives and (hopefully) make Odoo fit in better with the workflow of the company. However, before you attempt to write custom Odoo modules, it is important that you completely understand the functionality of Odoo and the various modules that are available in the community.

In this chapter, we will cover the following topics:

- Learning the basic structure of an Odoo module
- Using a module to add additional fields to your Odoo system
- Extending the views in your Odoo instance to include new fields
- Making changes to the available states to use in an Odoo workflow

Through careful configuration, many business objectives can be achieved without writing custom modules. Before you go down the path of writing custom Odoo modules, it is important that you are absolutely sure that your business requirements are clear and you have thoroughly explored all of the options available in Odoo. There are many settings that provide additional functionality to the Odoo system. You don't want to spend days, weeks, or even months building an Odoo module to then find out that much of that functionality was already available.

The goal of this chapter is to introduce you to custom module development in Odoo. Even if you are a beginner and don't know much about programming, you should be able to follow along and build a module in Odoo. If you don't know Python or XML, you are likely to find some aspects of this chapter a little more challenging. Fortunately, there are many resources in the Odoo community that can help you along your path to Odoo development.

Exploring the Odoo application and module directory

In addition to the built-in Odoo modules and the various settings that can change the way Odoo functions, there is also a growing collection of custom Odoo modules written by the community. When you find a business requirement for which you believe you may need to do some custom module development, take the time to go to the Odoo application repository and search for modules that could perhaps fit the purpose. Even if the module is not exactly what you are looking for, there can often be a lot of valuable code in those modules that can help you with your own module development.



You can find the Odoo application and module repository at
<https://www.odoo.com/apps>.

Even more importantly, these applications are invaluable for studying how to build and customize applications in Odoo. When you get stuck trying to solve a development problem, the Odoo source code and the available applications that you can find in the Odoo application repository are often all that you need to find the solution.

In Odoo development, we can sometimes refer to custom applications and modules interchangeably. For the most part, modules are customizations that extend Odoo applications with additional functionality. Odoo applications typically add an entire new set of features that are more substantial than a module. There is, however, no real difference between them in how you approach development. Both are created in the same way.

Building our first Odoo module

One of the best features of the Odoo framework is that we can extend Odoo and write our own modules without having to modify any of the Odoo source code. Instead, the changes we make are all contained in their own directory and their own files.

The primary advantage of this is that, when Odoo modifies their source code with patches or bug fixes, we do not have to worry about our changes getting overwritten. Also, while we may still need to modify our code if Odoo makes a dramatic change to their source code, there is a reasonable chance that the changes required will be minimal.



WARNING: Like in other areas of Odoo development, you should make frequent backups of your databases. Some of the examples we will show make changes to the database that can be difficult to undo.

Each module in Odoo has some basic requirements for it to be properly recognized by the Odoo framework and then installed. Once we successfully install our module, the framework will extend Odoo with the appropriate functionality.

Preparing your basic development environment

Before you can begin creating an application, you must make preparations on the server on which you wish to develop your module. This includes having an installation of Odoo that you can use just for development. You should not be doing any development on a production server. For this example, we are going to assume that you have followed the instructions for installing Odoo on an Ubuntu server, as was outlined in [Chapter 1, Setting Up Odoo 12](#).

Finding your Odoo configuration file and installing it

During your installation of Odoo, a configuration file was created that contains the necessary parameters for starting your Odoo server.

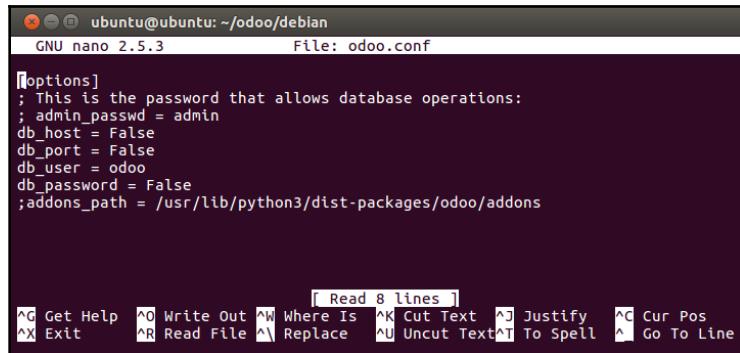


In the Ubuntu desktop, you can use the key combination *Ctrl + Shift + T* to bring up a Terminal window.

If you have performed a manual installation, then this file is in the `Debian` directory. Alternatively, if you have used a different installation from the Git repository or other sources, you may find the `odoo.conf` file in the `Debian` folder of your source installation. We will now look at the Odoo configuration file so we can find the directory for our Odoo installation and modify `addons_path` to contain our new directory. Open it in the Nano text editor using the following command:

```
sudo nano /etc/odoo/odoo.conf
```

Look at the following screenshot:



The screenshot shows a terminal window titled "ubuntu@ubuntu: ~/odoo/debian" running the "GNU nano 2.5.3" text editor. The file being edited is "File: odoo.conf". The content of the file is as follows:

```
[options]
; This is the password that allows database operations:
; admin_passwd = admin
db_host = False
db_port = False
db_user = odoo
db_password = False
addons_path = /usr/lib/python3/dist-packages/odoo addons
```

At the bottom of the screen, there is a menu bar with various keyboard shortcut commands. Some of the visible commands include: Get Help, Write Out, Where Is, Cut Text, Justify, Cur Pos, Exit, Read File, Replace, Uncut Text, To Spell, and Go To Line.

You will notice that the very last line in our configuration file contains `addons_path` to the location of the source of the Odoo applications.

Nano is a relatively simple text editor. A few of the more important commands you will need to use include the following:

- *Ctrl + O* is used to write out any changes you may make. You will be prompted for a filename.
- *Ctrl + X* is used to exit Nano.
- *Ctrl + C* is used to cancel an action.

After exiting Nano, you can verify the location of the Odoo applications by using the following commands in the Terminal:

```
cd /usr/lib/python3/dist-packages/odoo addons
```

If you use the `ls` command, you will then see the directories containing the source code for the Odoo applications.

Specifying a custom directory to hold our Odoo module

We will begin by creating a directory to hold our Odoo module. We have two options where we can create the directory to hold our module. For our first option, we could create our directory in the `addons` folder where all of the rest of the add-ons for Odoo are stored. This method is easy and allows Odoo to see our module simply by restarting the Odoo server.

A more preferable method, and the one we will use for this example, is to create a separate folder to hold our add-ons. This method has the advantage of keeping our modules separate from the standard Odoo modules.

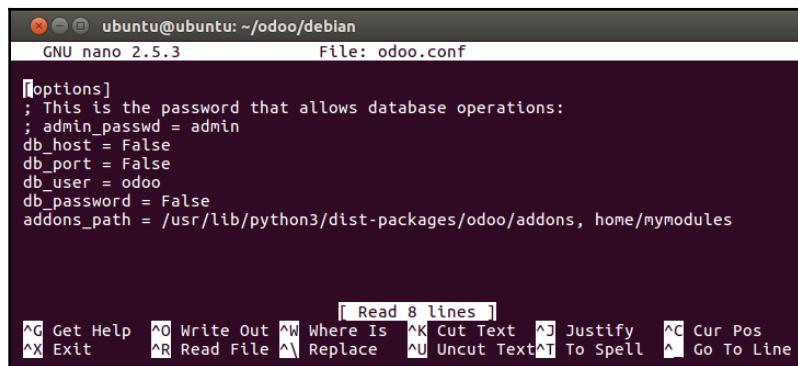
Create a new directory to hold our modules by typing the following command in the Terminal:

```
sudo mkdir /home/mymodules
```

For Odoo to find this directory, we need to modify the Odoo configuration file that we looked at previously:

```
sudo nano /etc/odoo/odoo.conf
```

Here, we can see a screenshot of the configuration file:



```
ubuntu@ubuntu: ~/odoo/debian
GNU nano 2.5.3          File: odoo.conf

[options]
; This is the password that allows database operations:
; admin_passwd = admin
db_host = False
db_port = False
db_user = odoouser
db_password = False
addons_path = /usr/lib/python3/dist-packages/odoo/addons, home/mymodules

[Read 8 lines]

[G] Get Help [W] Write Out [W] Where Is [C] Cut Text [J] Justify [C] Cur Pos
[X] Exit [R] Read File [R] Replace [U] Uncut Text [T] To Spell [G] Go To Line
```

We have edited the `odoo.conf` file to contain the `mymodules` directory within our `home` directory to store our custom Odoo modules.

Now, we need to create a directory to hold the actual module itself:

```
sudo mkdir /home/mymodules/silkworm
```

When you are getting started, it is important that you understand how to create these required files and how they work together. Depending on the build of Odoo that you have installed, you can use scaffolding to create an Odoo application more quickly. While the templating can save time, how it is used has undergone a lot of alteration, and the syntax for using it changes frequently. It is therefore recommended that you always keep yourself aware of the techniques of how to create an Odoo application without relying on scaffolding.



Contents of your module directory

Within our module directory, `silkworm`, we will create two files that are required in every Odoo module.

To navigate to our module directory, use the following command:

```
cd /home/mymodules/silkworm
```

These two files must always be named the following:

- `__init__.py`
- `__manifest__.py`

Although it is difficult to tell, in both cases there are two underscores together at the beginning and then another two underscores just before the file extension. You must name these files exactly this way in order to have a valid Odoo module.

These are Python files, and they can be edited with any text editor. We will begin by defining these two required files.

Creating and editing the files

Depending on your operating system of choice, there are a variety of editors you could use to create and edit the files for your module. In Windows, you could use something as simple as Notepad. In Ubuntu, there are also several choices, including Nano, Vi, or Vim.



Most developers will use a more full featured editor, such as Microsoft Developer Studio on the Windows platform or PyCharm on Ubuntu.

The `__init__.py` file

The purpose of the `__init__.py` file is to specify the Python files you wish to include in your module. At a minimum, you will usually have one Python file, but you could have more or less depending on the complexity of the module you are developing. If you were to have a Python file with your code, and that file was named `codexample.py`, you would have to import `codexample` inside the `__init__.py` file. You will notice that you don't have to include the `.py` extension inside the `__init__.py` file.

To create the `__init__.py` file in Ubuntu, make sure you are in the module directory and use the following command:

```
sudo nano __init__.py
```

This will bring up the blank text editor.

For our example, the `__init__.py` file will contain one line that specifies the name of the file in which we will be placing the Python code for our module:

```
import silkworm
```

The `__manifest__.py` file

The `__manifest__.py` file is essentially a manifest for your Odoo module. It describes the necessary attributes of your module to the Odoo framework. Sometimes, this file is also called the module descriptor file.

To create the `__manifest__.py` file in Ubuntu, make sure you are in the `module` directory and use the following command:

```
sudo nano __manifest__.py
```

The structure in the file is what is called a **dictionary** in Python:

```
{
    'name': 'Screen Printing',
    'version': '1.0',
    'description': """
        This module adds functionality for
        screen printing companies
    """,
    'author': 'Greg Moss',
    'depends': ['base', 'sale'],
    'data': ['silkworm_view.xml'],
    'demo': [],
    'installable': True,
    'auto_install': False,
}
```

This is how the `__manifest__.py` file appears when edited in Nano.

The `__manifest__.py` file contains a single Python dictionary. Even if you don't know Python, the syntax is rather simple if you have had even a little experience in programming. When you install a module in Odoo, this file describes the details the framework needs to properly configure your module, which are as follows:

- `name`: The `name` entry is what will appear in the module's listing inside Odoo.
- `version`: This allows you to specify a version number for your module. This will be valuable as you extend the functionality of your module, as you will need to keep track of the various releases.
- `description`: This `description` will appear when you prepare to install the module in Odoo. It should clearly describe the purpose of the module to someone who may be entirely unfamiliar with it. You should take the time to fill out this entry. Even this little bit of documentation can help someone who is trying to utilize the module in the future.



In the preceding `__manifest__.py` code, note the triple double-quotes before and after the `description` value. Python uses this syntax to allow you to continue a string on multiple lines.

- `author`: Providing the name of the `author` of your module is also important, as it could help future users to track down the main person who can provide assistance.
- `depends`: The preceding elements were pretty self-explanatory and are mostly for documentation purposes. This entry, however, tells the framework what other modules your module will build upon. At a minimum, you will need to include `base` as one of your module dependencies. In our example, we will be extending the sales order system, so we have also included `sale` as one of the module dependencies.
- `data`: The `data` item specifies the XML view files you wish to include in your module. We will cover view files in depth later in this chapter. If you plan to change something in Odoo's forms or user interface, it will most likely involve creating a view file. Other types of data files can be specified here, such as files containing initial data or access rights, but for our example, we have named only the `silkworm_view.xml` file.
- `demo`: Odoo provides a rather convenient method of including demonstration data with your module. When you create your database, you have the option to include demonstration data with that Odoo instance. We have left this blank for our example, but if we wished to make demonstration data available when the module is installed, we could fill in this entry.

- `installable`: This is an entry that you may use to temporarily disable a module for installation. Most often it will be `True` because you want the ability to install the module in an Odoo instance.
- `auto_install`: When this entry is set to `True`, Odoo will automatically install this module when it finds that all of the dependency modules are installed. If you have no dependencies, this means that it will be automatically installed when you create a new database. Given Odoo's modular application approach, you typically would not want to have the `auto_install` flag set to `True` for most module development.

Extending an Odoo model in `silkworm.py`

Next, we create another file named `silkworm.py`. We will begin by creating a module that performs the same customization we performed through developer mode in Chapter 13, *Customizing Odoo for Your Business*.

Why would we want to put our customization into a module rather than just using developer mode?

First off, changes made through developer mode are isolated within that instance of Odoo. If you decide you wish to create a new database, you will have to make all of the developer changes by hand, again. More importantly, when you make the changes in a module, you have much more control over the final results.



Developer mode is very powerful for quickly looking at views, analyzing fields on forms, and understanding more about the Odoo framework. However, it is typically far better to make any actual changes by creating a module rather than modifying the views or models in developer mode. So, unless you are using on Odoo Online, you are better to avoid changes in developer mode.

Using a module to add custom fields to a model

In Chapter 10, *Creating Advanced Searches and Dashboards*, we added **Date Required** and **Rush Order** to our sales order model. Now, let's see how we can do the exact same thing in our module.

In our `__init__.py` file, we only had one line: the `import silkworm` command.

To add the **Date Required** and **Rush Order** fields to our sales order, we can place the following in the `silkworm.py` file:

```
from . import models, fields

class silkworm_sale_order(models.Model):
    _inherit = 'sale.order'

    daterequired = fields.Date('Date Required')
    rush = fields.Boolean('Rush Order')
```

Please note that these are not custom fields created in developer mode preceded by `x_`. In Python, the `from` command allows you to specify which libraries you wish to utilize in your custom classes. For our simple example, we are only pulling in `models` and `fields`.

Inheriting from the Odoo Sales application

In our `class` statement, we specify the `silkworm_sale_order` class, and it has the `models.Model` parameter. Remember that when learning the Odoo framework, it will take a bit of time to get familiar with the syntax. Right now, you don't necessarily have to understand why you are specifying `models.Model`; you just need to understand that it is required with most classes. Refer to the following code snippet as follows:

```
_inherit= 'sale.order'
```

For those who are new to object-oriented programming in general, the `_inherit` statement essentially makes the functionality of the Odoo sales application available to your class so you can extend it with your own fields and methods.

Next, we can extend the Odoo sales application with our two custom fields, as shown in the following:

```
daterequired = fields.Date('Date Required', required=True)
rush = fields.Boolean('Rush Order')
```

You will note that, in the syntax, we also specify the data types and provide the labels we want to display in the views inside Odoo. Note that we have also set `required=True` for the `daterequired` field so that the user will be forced to provide this data when they create a sales order record if that field is contained in the view.

Understanding Python conventions

Unlike many programming languages, Python takes white space very seriously. In fact, you must indent your code exactly or the Python compiler will generate an error. For example, the `_inherit` attribute is indented exactly four spaces over from the `class` command.

Adding the fields to our sales order view

Now that we have specified the fields that we want added to our sales order model, we must create our view file that will display the fields in the sales order header. We have specified the name of this file inside of `__manifest__.py` within the `data` entry. For our example, the filename is `silkworm_view.xml`.

Using your editor of choice, create the `silkworm_view.xml` file. In this file, enter the following code:

```
<?xml version="1.0" encoding="utf-8"?>

<Odoo>
    <data>
        <record id="sale_view_order_form" model="ir.ui.view">
            <field name="model">sale.order</field>
            <field name="inherit_id" ref="sale.view_order_form"/>
            <field name="arch" type="xml">
                <field name="payment_term_id" position="after">
                    <field name="daterequired"/>
                    <field name="rush"/>
                </field>
            </field>
        </record>
    </data>
</Odoo>
```

Now, let's walk through this code and describe what it does. Odoo specifies views using XML syntax. The first line in the file is the standard element you will find at the top of many XML files, specifying the version and type of encoding used.

The next thing to bear in mind is that Odoo view files contain beginning and ending Odoo tags. Inside those tags, there are matching opening and closing `data` tags. To modify or add views in your custom Odoo module, you must add `record` tags.

Each record must have an `id`. In this case, we also have a `model` tag that is specified as `ir.ui.view`:

```
<record id="sale_view_order_form" model="ir.ui.view">
```

This is a framework convention, and you will learn about other models that are available as you continue to study Odoo development.

Next, we must specify the base `model` with which this view interacts. For our example, this is `sale.order`. This relates directly to the fact that we have added the fields to the `sale.order` model in our Python file:

```
<field name="model">sale.order</field>
```

If instead you were adding additional fields to the purchase order header, you would specify `purchase.order`.



Use developer mode in Odoo to mouse over fields and determine which models they relate to. To find the view names which you need to use within your code, go into **Manage Views** in developer mode. This can save you a great deal of time when developing in Odoo.

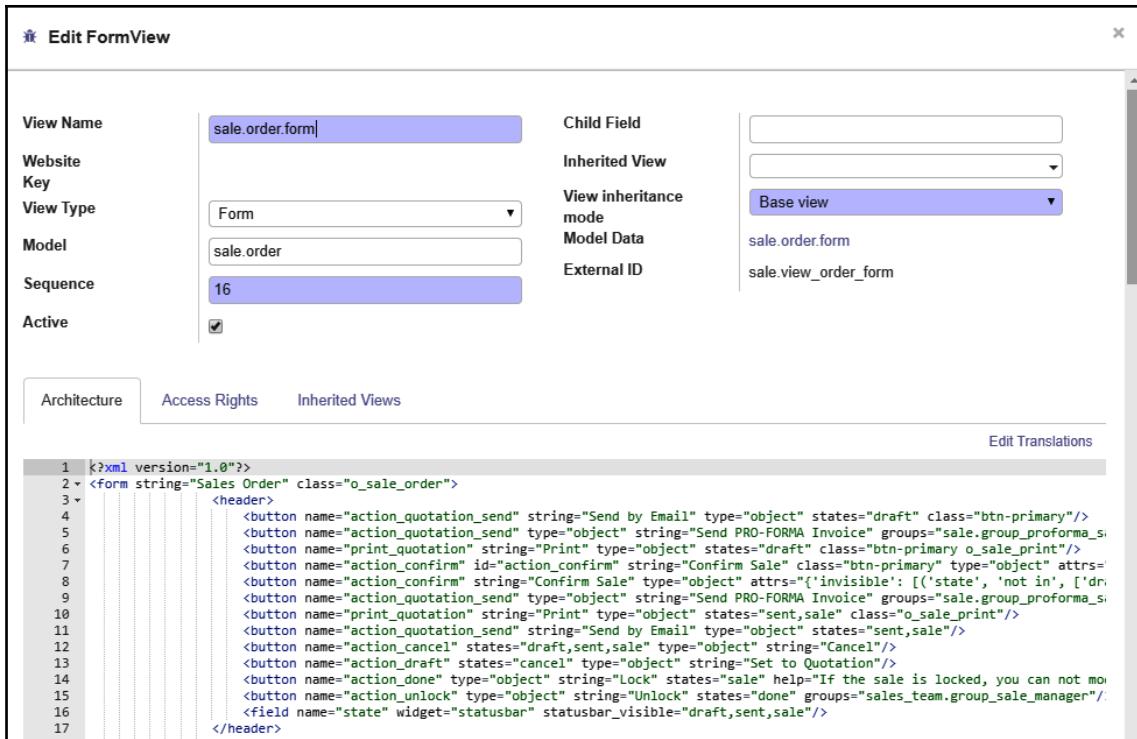
Next, let's look at the line that contains `inherit_id`:

```
<field name="inherit_id" ref="sale.view_order_form"/>
```

In much the same way that we had to inherit from `sale.model` when we created our `silkworm_sale_order` class, we must inherit from the `sale.view_order_form` view so that we can add the additional fields.

How did we know that we had to inherit from `sale.view_order_form`? One big trick for finding the value you require is to use the edit form view while in developer mode.

For this example, while on a sales order in Odoo, choose **Edit Form View** from the Developer Tools menu. You will be then taken to the form that shows you the exact **External ID** you need to add fields to the form as shown in the following screenshot:



The screenshot shows the 'Edit FormView' window for a view named 'sale.order.form'. The configuration includes:

- View Name:** sale.order.form
- Website Key:** (empty)
- View Type:** Form
- Model:** sale.order
- Sequence:** 16
- Active:** checked
- Child Field:** (empty)
- Inherited View mode:** (empty)
- View inheritance mode:** Base view
- Model Data:** sale.order.form
- External ID:** sale.view_order_form

Below the configuration, there are tabs for Architecture, Access Rights, Inherited Views, and Edit Translations. The XML code for the form view is displayed in the Architecture tab:

```

1 <?xml version="1.0"?>
2 <form string="Sales Order" class="o_sale_order">
3 <header>
4   <button name="action_quotation_send" string="Send by Email" type="object" states="draft" class="btn-primary"/>
5   <button name="action_quotation_send" type="object" string="Send PRO-FORMA Invoice" groups="sale.group_proforma_s...
6   <button name="print_quotation" string="Print" type="object" states="draft" class="btn-primary o_sale_print"/>
7   <button name="action_confirm" id="action_confirm" string="Confirm Sale" class="btn-primary" type="object" attrs='{"invisible": [{"state": "not_in", ["dr...}]}
8   <button name="action_confirm" string="Confirm Sale" type="object" attrs='{"invisible": [{"state": "not_in, ["dr...}]}
9   <button name="action_quotation_send" type="object" string="Send PRO-FORMA Invoice" groups="sale.group_proforma_s...
10  <button name="print_quotation" string="Print" type="object" states="sent,sale" class="o_sale_print"/>
11  <button name="action_quotation_send" string="Send by Email" type="object" states="sent,sale"/>
12  <button name="action_cancel" states="draft,sent,sale" type="object" string="Cancel"/>
13  <button name="action_draft" states="cancel" type="object" string="Set to Quotation"/>
14  <button name="action_done" type="object" string="Lock" states="sale" help="If the sale is locked, you can not mo..."/>
15  <button name="action_unlock" type="object" string="Unlock" states="done" groups="sales_team.group_sale_manager"/>
16  <field name="state" widget="statusbar" statusbar_visible="draft,sent,sale"/>
17 </header>

```

The **Edit FormView** screen shown here shows the **View Name** that has been assigned to the view we are currently looking at. Now, we know that the **External ID** we need to use is `sale.view_order_form`.



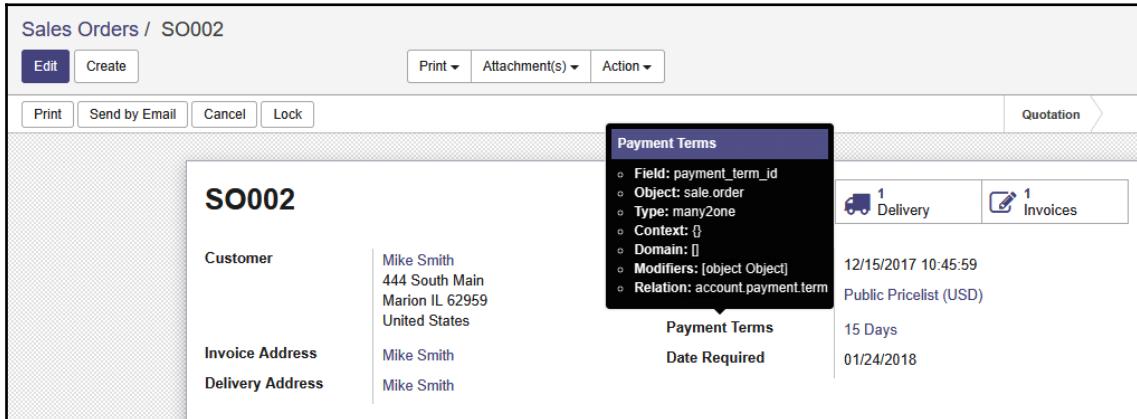
The **External ID** is a unique identifier for the view you are working on. This is how the Odoo framework knows which view you are referring to.

When you are adding fields to a form, it is important for Odoo to have the information it requires to determine exactly where the fields should go. In this example, we are telling the Odoo framework we want to first find the field named `payment_term_id`. Next, we use `position="after"` to specify that we want the fields to appear after `payment_term_id`.



In addition to `after`, the `position` attribute can be specified as `position="before"` to place a field before that element or `position="replace"` to replace an element. So, if you'd used `replace` instead of `after`, the `payment_term_id` field would be replaced by new fields that we add as follows.

Once again, we can use developer mode to visually find the field name we require. In the following screenshot, we have moved the mouse over the **Payment Terms** field to reveal details about that field:



Now that we know where to add our fields, we can specify custom fields to display:

```
<field name="daterequired"/>
<field name="rush"/>
```

Getting ready to install our module

Right now, our module is very simple and just adds two fields to our sales order form. We should still quickly review the following files that you should have in your `module` directory:

- `__init__.py`
- `__Odoo__.py`
- `silkworm.py`
- `silkworm_view.py`

When you run Odoo in Ubuntu, it is good practice to run the service under a special account that has limited permissions. This is set up automatically when you use the Debian install. Therefore, we need to change the permissions on our `module` directory so that Odoo can properly access the files. Use the following command to set the permissions:

```
sudo chown odoo:odoo /home/mymodules -R
```

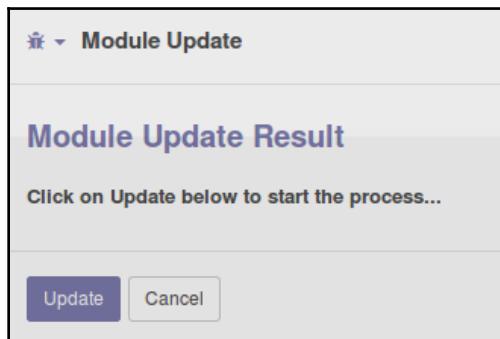
To install the module, you must also restart the Odoo server. If you don't restart your server, then Odoo will not see your module. Enter the following command:

```
sudo /etc/init.d/odoo stop
```

Then, enter the following command:

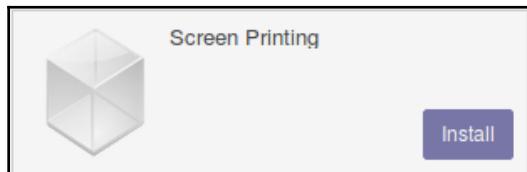
```
sudo /etc/init.d/odoo start
```

In the top menu, click on **Apps** and choose **Update Apps List** from the menus at the top. You will see the following screenshot:



On the **Module Update** screen, click on the **Update** button. Once you've clicked **Update**, Odoo will refresh the available list of add-ons.

Next, we will install the module. Remove the **Apps** filter from the search box (yes, this entire process is somewhat counter-intuitive). Once you remove the **Apps** filter, you can search for `silkworm` to locate your new module for installation. Note that the application is shown here as **Screen Printing** because that was the name we assigned to it within our `__manifest__.py` code file:



You do not need to give your application and its directory the same name. This allows you to keep the directory name short while giving the application a longer descriptive name.

Click the **Install** button to begin the installation process. After a few seconds, the screen will refresh. You can now pull up a sales order and see the fields added to your form shown as follows:

The screenshot shows a sales order interface with the reference number SO001 at the top. On the left, there's a sidebar with tabs for 'Order Lines' and 'Other Information'. Under 'Order Lines', a single item is listed: 'Medium White T-Shirt' with a quantity of 1.000, unit price of 16.50, and taxes applied (Tax 15.00%). The subtotal is \$ 16.50. Below this, there's a link to 'Add an item'. On the right side, there are several configuration fields: 'Customer' set to 'Mike Smith', 'Expiration Date' set to '02/15/2018', 'Payment Terms' set to '15 Days', 'Date Required' set to '02/01/2018', and a checked 'Rush Order' checkbox. At the bottom, there's a note about setup default terms and conditions, followed by a breakdown of the total amount: Untaxed Amount: \$ 16.50, Taxes: \$ 2.48, and Total: \$ 18.98.

This screenshot of a sales order shows the custom fields added in our module.



When developing, it is sometimes inevitable that a module may not install correctly, or that, after installing, you have an error that will prevent you from logging in to Odoo. If you find yourself unable to resolve the error, one workaround to get Odoo back up and running again is to rename the module directory. This prevents Odoo from locating the module to install.

Troubleshooting your module installation

You won't be the first Odoo developer who created a module from scratch to not have it show up in the list of apps. There are a few things you can check if you don't see your application in the list.

First of all, it is always good to know how to check the log file for errors in your Odoo installation. This is particularly important while you are developing an Odoo module. If you followed the standard Ubuntu Debian installation, you can open the log file and view it with the following command:

```
sudo nano /var/log/odoo/odoo-server.log
```

To demonstrate how this can be useful, we have modified the `silkworm_view.xml` file to contain a small error. When we attempt to install the module, we get an error that we can then view in the `odoo-server.log` file:

```
adminuser@ubuntu: /home/mymodules/silkworm
GNU nano 2.5.3                               File: /var/log/odoo/odoo-server.log

doc = etree.parse(xmlfile)
File "/src/lxml/lxml.etree.pyx", line 3427, in lxml.etree.parse (src/lxml/lxml.etree.c:85131)
File "/src/lxml/parser.pxi", line 1803, in lxml.etree._parseDocument (src/lxml/lxml.etree.c:124287)
File "/src/lxml/parser.pxi", line 1823, in lxml.etree._parseFilelikeDocument (src/lxml/lxml.etree.c:124599)
File "/src/lxml/parser.pxi", line 1718, in lxml.etree._parseDocFromFilelike (src/lxml/lxml.etree.c:123258)
File "/src/lxml/parser.pxi", line 1139, in lxml.etree._BaseParser._parseDocFromFilelike (src/lxml/lxml.etree.c:124599)
File "/src/lxml/parser.pxi", line 573, in lxml.etree.ParserContext._handleParseResultDoc (src/lxml/lxml.etree.c:112276)
File "/src/lxml/parser.pxi", line 683, in lxml.etree._handleParseResult (src/lxml/lxml.etree.c:112276)
File "/src/lxml/parser.pxi", line 613, in lxml.etree._raiseParseError (src/lxml/lxml.etree.c:111124)
XMLSyntaxError: expected '>', line 13, column 10
2017-01-12 22:37:26,851 20943 ERROR SILKWORM-DEV odoo.http: Exception during JSON request handling.
Traceback (most recent call last):
  File "/usr/lib/python2.7/dist-packages/odoo/http.py", line 638, in _handle_exception
    return super(JsonRequest, self)._handle_exception(exception)
  File "/usr/lib/python2.7/dist-packages/odoo/http.py", line 675, in dispatch
    result = self._call_function(**self.params)
  File "/usr/lib/python2.7/dist-packages/odoo/http.py", line 331, in _call_function
    return checked_call(self.db, *args, **kargs)
  File "/usr/lib/python2.7/dist-packages/odoo/service/model.py", line 119, in wrapper
    return f(dbname, *args, **kargs)
  File "/usr/lib/python2.7/dist-packages/odoo/http.py", line 324, in checked_call
    self.db.cursor().execute(query)

Get Help   Write Out   Where Is   Cut Text   Justify   Cur Pos   Prev Page
Exit      Read File   Replace   Uncut Text To Spell  Go To Line  Next Page
```

When we look at the error log, we can see `XMLSyntaxError`, which was introduced by the error we put into the `silkworm_view.xml` file. The error log is your first stop in attempting to solve these problems.

Now that you know how to identify problems in an installation, here are some typical things to check for that could be preventing your module from installing properly:

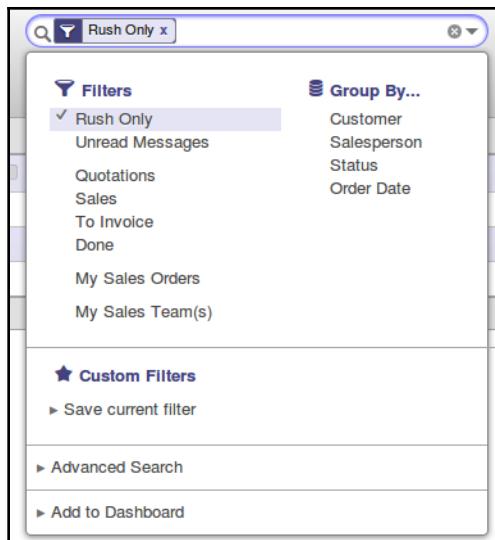
- Verify that the permissions are set correctly on your files. You can use `ls -l` to list out the files with their permissions.
- Make sure that the add-on path for your module directory is in the configuration file. You can verify that it is set by examining the log file. It should be one of the first lines in the log file after restarting the server.
- Stop and restart the Odoo server.

- Check whether your `__manifest__.py` file is missing, is named incorrectly, or has any typos.
- Verify that you have no indentation problems within your Python files.
- Sometimes, it can help to refresh your browser after installing a new application.
- Always start simple. When in doubt, test out your models first, then add in the view files once you have verified that the models have been modified as you expected.

Using a module to add a filter to a search view

One very nice feature of Odoo is the flexible but easy-to-use search functionality that is provided on every list view. With a module, you can add additional filter options that make it easier for users to find the information they are looking for. In our real-world example, we have placed an importance on rush orders. Therefore, it would be desirable to have a filter option on our sales order view that will limit our listing to only display rush orders.

Here is how the final search filter view will appear after we implement the module changes:



When **Rush Only** is checked, the sales order list view will limit the orders to only those orders that are specified as rush orders. Users can now quickly locate rush orders without creating custom filters. This is an example of how a small change can have significant real-world benefits to usability. Best of all, with Odoo, you are making these changes without modifying any of the base Odoo source code.

Adding the code to create the rush order filter

The code segment will naturally be added to the `silkworm_view.xml` file. It will have the same record structure as our other modification. Adding this code segment and updating the module will implement the change we want:

```
<record id="sale_view_sales_order_filter" model="ir.ui.view">
    <field name="name">sale.order.search</field>
    <field name="model">sale.order</field>
    <field name="inherit_id" ref="sale.view_sales_order_filter"/>
    <field name="arch" type="xml">
        <field name="name" position="after">
            <filter name="rush" string="Rush Only"
                domain="[( 'rush', '=', True )]"/>
        </field>
    </field>
</record>
```

Let's look at some of the more important elements of this code segment. It follows a similar structure as the modification that added fields to our form. What is most important when looking at the code in any module is to identify the `inherit_id` field's `ref` value. This is what ties your view modifications to the view in the base module.

In this case, `inherit_id` is `sale.view_sales_order_filter`.



Use developer mode to look up the view name from inside Odoo. Navigate to the view you want to work with and, in the Developer Tools menu, you can choose **Edit Action** to see the **External ID** of the view. You can also use developer mode to quickly look at the syntax of views and use them to help you to determine how your filters should be structured.

Creating the filter

The filter is specified by the following one line of XML code:

```
<filter name="rush" string="Rush Only"  
domain="['rush', '=', True]"/>
```

In this code, we specify the name of our filter and the string we wish to display in the search view. The filter is applied with the domain parameter. We specify the field from our sales order model and that it must equal `True` in order for this filter to be valid.

The technical name for this syntax in Python is a **tuple**. It is possible to include multiple filters in the domain. For example, we can also specify that we only want sales orders that are confirmed by specifying an additional condition in our filter:

```
<filter name="rush" string="Rush Only"  
domain="['rush', '=', True], ('state', '=', 'progress')"/>
```

Odoo considers a confirmed sales order to be in a state named `progress`. With this change, our `Rush Only` filter will also limit the sales orders to only those that are confirmed.

Using Odoo to create websites and web services

In the previous examples, we have been extending the Odoo framework to include additional fields in models and functionality in our views. Odoo also provides a powerful framework for creating your own websites and web services that can integrate easily with Odoo applications.

Let's see how we can create a simple web service that displays the rush orders on a page.

We begin by creating a controller that is tied to a URL. When we navigate to this URL in our browser, the controller will do whatever processing we require.

Create the controller file using the following command in your Terminal window:

```
sudo nano controller.py
```

To test our controller and make sure it is functioning properly, place the following code in the `controller.py` file to create a simple output:

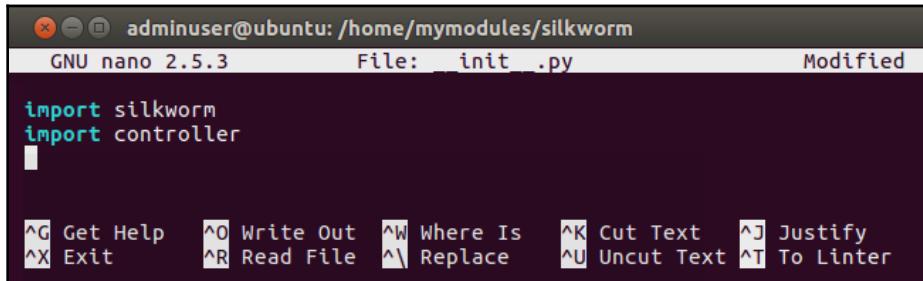
```
from odoo import http

class Web_RushOrders(http.Controller):
    @http.route('/orders/rush/', auth='public')
    def index(self, **kw):
        return "Rush Orders"
```

You must now edit the `__init__.py` file to include the new controller file:

```
sudo nano __init__.py
```

Your Python file will open within the Nano text editor. Append it with these two `import` commands and save your change:

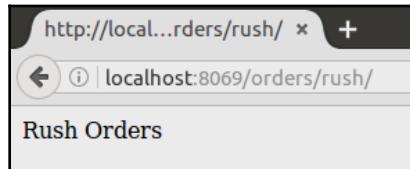


```
import silkworm
import controller
```

This allows the `controller.py` file we have created to get picked up by the Odoo framework.

Now, stop and start the Odoo server and navigate to `http://localhost:8069/orders/rush`.

You will then see the following custom web page:



Now that we have tested our controller and have a very simple page, let's see how we can hook into the Odoo sales application and display our rush orders.

Creating an XML template

For testing purposes, it was fine to output some text to our web page to verify that our controller is working. When you are designing a full website, however, this can be very tedious, and it is considered bad practice to mix your Python code and logic with your HTML code that you will use to display your website. Templates allow us to more easily separate the programming logic of our applications from the information we display.

Create a file to hold the template we will use to display our rush orders:

```
sudo nano template.xml
```

Enter the following code into the editor:

```
<odoo>
    <data>
        <template id="index">
            <title>Rush Orders</title>
            <table>
                <t t-foreach="rushorders" t-as="rushorder">
                    <tr>
                        <td><t t-esc="rushorder.name"/></td>
                        <td><t t-esc="rushorder.daterequired"/></td>
                    </tr>
                </t>
            </table>
        </template>
    </data>
</odoo>
```

You will see that we are using XML in much the same way that we have used in earlier examples. The HTML code used is simple, and we have only included the `name` (for example, `SO001`, and `SO002`) and `daterequired` fields.

Most significantly, you will see that we are using a `foreach` loop that will go through the orders and display the `name` field in one column and then the `daterequired` field in another column.

Now, we need to update our controller to pass along sales order data to the template.

Open up the `controller.py` file and change the contents to the following:

```
from odoo import http

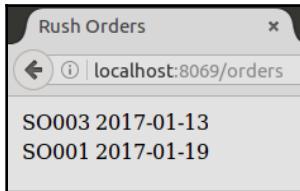
class Webrushorders(http.Controller):

    @http.route('/orders/rush/', auth='public')
    def index(self, **kw):
        Orders = http.request.env['sale.order']
        return http.request.render('silkworm.index', {'rushorders':
Orders.search([('rush','=',True)])})
```

Everything stayed the same until we got inside the `index` method. First, we define `Orders` to give us access to the `sale.order` model.

In our final line, we use the `http.request.render` method to access the template and pass along our orders. The `Orders.search` method is passed the filter that limits the order to only those in which the `rush` flag is `True`.

When we save our `controller.py` file, stop and restart the server and then update the module. We can see the following results in the web browser:



Make sure you flag a few orders in your system as `rush` or you will get an empty page.

Integrating with the Odoo API

Oftentimes when developing custom applications, you will create solutions that involve interoperability with other systems and platforms. For example, perhaps you need to integrate with a third-party **Customer Relationship Management (CRM)** application to create records inside of Odoo. The API is also quite useful for data migration.

Connecting to the API

Accessing the API is relatively easy. We begin with the code that imports the required libraries and creates a connection to the Odoo server:

```
import xmlrpclib
url = http://localhost:8069
db = SILK-DEV
username = 'admin'
password = 'admin'
info =
    xmlrpclib.ServerProxy('https://localhost:8089/start').start()
url, db, username, password = \
    info['host'], info['database'], info['user'], info['password']
```

Filtering and returning records through the API

We can use the same domain filters that we used in filtering rush orders, in order to use the API to return a list of sale order IDs that match as shown in the following snippet:

```
models.execute_kw(db, uid, password,
    'sale.order', 'search',
    [ ["[('x_rush', '=', True), ('state', '=', 'progress')]]])
```

Using the search_read method

While the preceding code only returns the IDs of the records, the new Odoo API allows you to both search and read the actual fields from the model with one single API call. Here, we return some fields from the sales order header as shown in the following snippet:

```
models.execute_kw(db, uid, password,
    'sale.order', 'search_read',
    [ [['x_rush', '=', True], ['state', '=', 'done']]],
    {'fields': ['name', 'country_id', 'comment'], 'limit': 5})
```

Creating custom themes in Odoo

Odoo offers a powerful integrated web builder application that you can use to publish your own business website. You can purchase themes from Odoo's app store, but in some instances, it is also highly desirable to build your own custom themes. This is a very advanced topic, but next, we will look at some of the basic steps to follow when building your own custom theme.

Basic architecture of Odoo themes

Odoo themes are more complicated than a simple set of HTML pages or a simple dynamic page that uses JavaScript. Odoo's theme framework is built around blocks, in which you can use drag and drop to assemble both your page structure and your page content. Before you start trying to create your own themes, make sure you are thoroughly comfortable with the basics of creating pages in Odoo's website builder.

If you are serious about developing themes, then you will need to understand the main technologies that make up the framework. These include Bootstrap, jQuery, jQuery UI, and underscore.js. Also, as with other Odoo development tasks, you will gain a lot of insight by examining the existing themes that are included with a basic Odoo installation.

Creating an Odoo theme using scaffolding

To create an Odoo theme, you basically use the same structure as any other Odoo module. You will need a main folder to hold your theme files. This folder should be named `theme_` followed by the name of your theme.

Fortunately, you can use scaffolding to create a template for your module so you don't have to create all of the files from scratch. To create our custom theme, use the following command in a Terminal window. Make sure you are in the directory that contains `odoo-bin` and that you use the correct name for your custom add-on directory shown as follows:

```
./odoo-bin scaffold -t theme "Silkworm Theme" custommodules
```

The scaffolding includes references to empty files inside the `__manifest__.py` file. Go in and clear out these references so your file looks like the following:

```
{  
    # Theme information  
    'name': "Silkworm Theme",  
    'description': """ A custom theme example
```

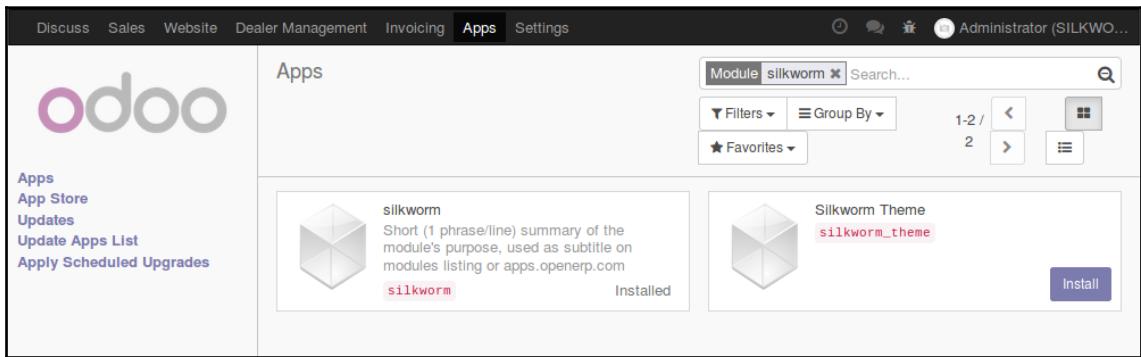
```
"""
'category': 'Theme/Creative',
'version': '1.0',
'depends': ['website'],

# templates
'data': [
],

# Your information
'author': "Your Name",
'website': "",
}
```

Specifically, we have removed references to the `options.xml` and `snippets.xml` files.

You can now restart the Odoo server and refresh the app list. The **Silkworm Theme** will now appear ready to install. You can now **Install** the theme and make sure there are no errors with the basic template as shown in the following screenshot:



Next, let's see how a `layout.xml` file is used to add new elements to the Odoo header.

Modifying the default Odoo theme header

When you install the basic Odoo theme, you will have a default Odoo header that has a navigation menu with the company logo. We can create an XML file that will extend the theme. Begin by creating a file named `Layout.xml` and placing it inside the `views` folder:

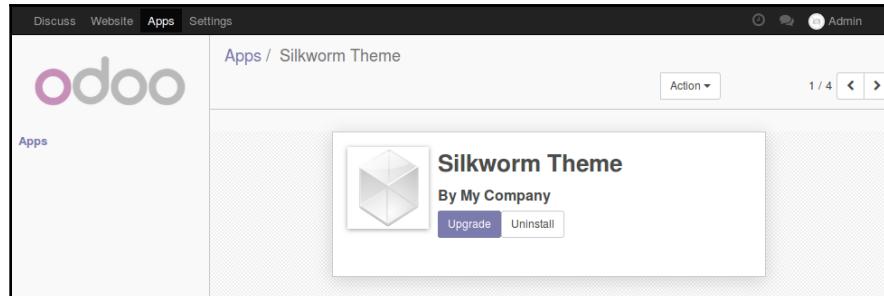
```
<odoo>
<data>
<template id="custom_header" inherit_id="website.layout" name="custom
Header">
```

```
<xpath expr="//div[@id='wrapwrap']/header" position="attributes">
<attribute name="id">custom_message</attribute>
</xpath>
<xpath expr="//div[@id='wrapwrap']/header/div" position="after">
<div class="container">
<div class="alert alert-info mt16" role="alert">
<strong>Welcome to Silkworm! </strong>
</div>
</div>
</xpath>
</template>
</data>
</odoo>
```

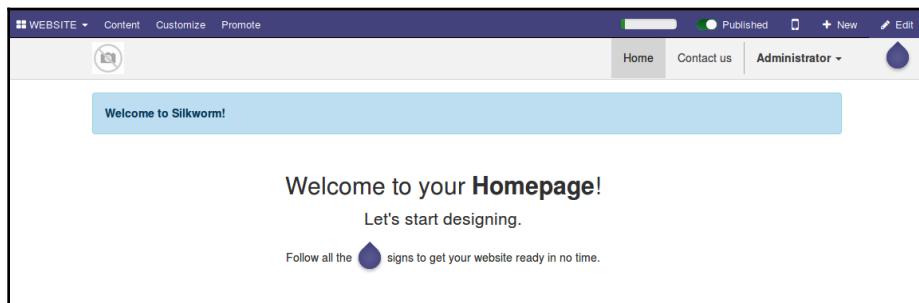
You will need to modify your `__manifest__.py` file data element to reference this new file:

```
'data': ['views/layouts.xml'],
```

To see your changes, you will need to restart the server and **Upgrade** the theme as shown in the following screenshot::



After you have upgraded the theme, you can navigate to any page on the website and see the change as shown in the following screenshot:



In the previous example, we learned how to modify the website template header that then modifies the look of every page on the website. Next, we will see how we can create a page that has a specific layout.

Creating a custom page layout

Let's say we want to create a unique page for art approval. We know that we want this to have a specific layout and look different to the rest of the site, so we create a specific layout page for it. In this example, we will just list a few recommendations for customers when approving the art. You can, however, extend this with any layout you choose.

We begin by creating a `pages.xml` file and pasting in the following code:

```
<odoo>

<record id="services_page" model="website.page">
    <field name="name">Art Approval</field>
    <field name="website_published">True</field>
    <field name="url">/approval</field>
    <field name="type">qweb</field>
    <field name="key">themeTutorial.artapproval_page</field>
    <field name="arch" type="xml">
        <t t-name="themeTutorial.artapproval_page_template">
            <t t-call="website.layout">
                <div id="wrap">
                    <div class="container">
                        <h1>Art Approval Checklist</h1>
                        <ul>
                            <li>Check the spelling of all proofs</li>
                            <li>Verify your color selections</li>
                            <li>Provide detailed feedback on any
                                changes</li>
                        </ul>
                        <!-- Area for Snippets === -->
                        <div class="oe_structure" />
                    </div>
                </div>
            </t>
        </t>
    </field>
</record>

</odoo>
```

Let

Let's talk a bit about some of the specific elements.

The `record` tag uniquely identifies our page.

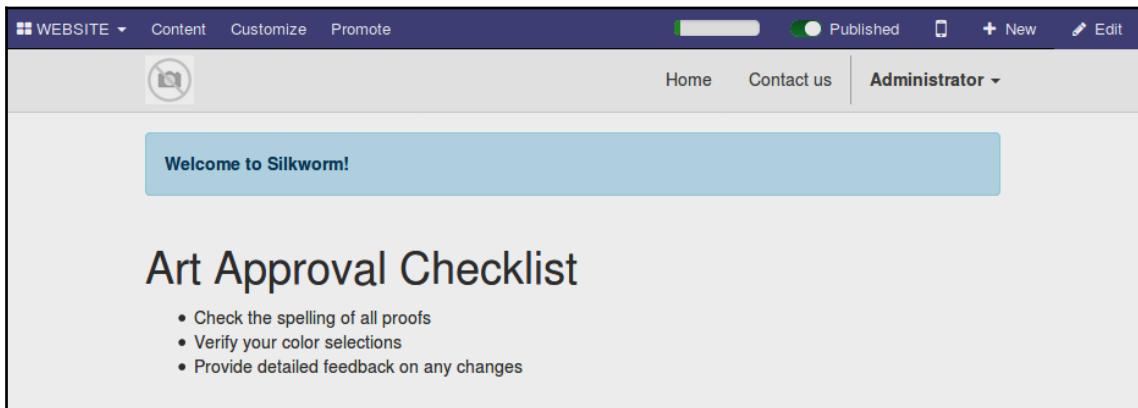
The `model` attribute is how Odoo knows that this entire record should be handled as a website page.

The `name` element provides a name to our page and flags it as being published. The `url` field specifies where the page resides in the site. You will use this to navigate to the page. The type will always be `qweb` as that is the template language Odoo uses for the pages. We then have a key that is used internally by Odoo.

Looking at the actual content of the page, you can see that we use a `t` element to specify our theme name and the call to `website.layout`. All of this then drills down to where we finally have the `div` tag with our container class. It is within here that we can specify the exact layout and default content for our page. In this case, we have given some simple instructions as an example.

Finally, we have a `div` class at the bottom with the `oe_structure` class. This is where we can now drag and drop blocks in the web builder in order to manage the content in the page.

Add the reference to `pages.xml` to your `__manifest__.py` file and upgrade the theme. If you are successful, the result should look similar to this:



This should give you a pretty good start at understanding theme development in Odoo. There is much more to learn for building snippets and your own blocks. Please take a look at the [Appendix A, Locating Additional Odoo Resources](#), section for additional resources on building your own theme.

Summary

In this chapter, we learned about the basic Odoo structure for modules. Files must be named in the exact way the Odoo framework expects, and you must follow the structure in order for your module to properly load into Odoo. We explored how to extend Odoo with additional fields and display them on forms. Next, we extended our custom module with a workflow example. This allowed us to see how we can peek into existing Odoo modules to assist us with developing our own module.

Next, we will look at the differences between Odoo Community Edition and Odoo Enterprise Edition. We'll learn to how to utilize the runbot to compare and contrast these two editions, as well as the various build versions of Odoo (12.0, 11.0, 10.0, and so on.). In the next chapter, we will also explore enhancements to Odoo 12.0, which include improvements to accounting reports, subscriptions, voice over IP integration, shipping connectors, and the new Product Lifecycle Management application.

16

Comparative Analysis of Community versus Enterprise Editions

Beginning with Odoo 9, and continuing through Odoo 12, the company Odoo S.A. started to offer a licensed version of Odoo known as **Odoo Enterprise Edition**. (Odoo S.A. is a Belgian company, and **S.A.** stands for **Société Anonyme**, which is the French equivalent to a corporation or "Inc.") This edition offers several enhancements over the free **Odoo Community Edition**, including an entirely redesigned user interface. In addition to actual enhancements to the software itself, the Enterprise Edition also includes version upgrades and bug fix guarantees that require Odoo SA to respond to and attempt to fix any known bugs you may encounter in the core Odoo applications.

In this chapter, we will cover the following topics:

- The primary differences between Odoo Community and Odoo Enterprise
- Improved accounting reports and external integration for the Enterprise Edition
- Sales management enhancements for **Voice over Internet Protocol (VoIP)** integration and subscriptions
- Enterprise shipping connectors
- Manufacturing applications, including **Product Lifecycle Management (PLM)**, maintenance, and quality

The primary goal of this chapter is to provide information that will help you to decide which version of Odoo, Community or Enterprise, is right for you.



There are many features and options in Odoo Enterprise that we will not have the opportunity to cover in just a single chapter. It is recommended that if you are considering Odoo, you should look at your specific requirements and research how Odoo Enterprise may be of benefit.

Getting an overview of the Community and Enterprise Editions

While there are a lot of significant—and not-so-significant—differences between Odoo Community and Odoo Enterprise, Odoo SA does provide a quick reference sheet that summarizes, at a very basic level, the differences between the two versions. You can find this quick reference at <https://www.odoo.com/page/editions>:

The screenshot shows a comparison table titled "Compare Odoo Editions". The table has two columns: "Enterprise" and "Community". It lists various features under categories like General, User interface, Studio, and Accounting. A checkmark (✓) indicates the feature is included in that edition, while a crossed-out checkmark (✗) indicates it is not.

	Enterprise	Community
General		
Unlimited Functional support	✓	✗
Version Upgrades	✓	✗
Hosting	✓	✗
User interface		
Desktop	✓	✓
Mobile Version	✓	✗
Studio		
Screen Customization	✓	✗
Report Designer	✓	✗
Menus Editor	✓	✗
Apps Creator	✓	✗
Accounting		
Invoicing & Payments	✓	✓

While the checklist does give you a high-level view of the differences between the various versions of Odoo, unfortunately, there is no explanation or additional details as to what each of these features provides. To learn more, we have to take a look at the Odoo Community and Enterprise Editions side by side.

Using the Odoo runbot to compare Odoo versions

One thing to consider when comparing the two editions of Odoo is that Odoo Enterprise requires that you pay license fees. At first, you may think that this makes it difficult to try Odoo Enterprise for yourself. Fortunately, there is a handy service, called **odoo runbot**, that will not only help you with comparing the two editions but can also help you to test out any version of Odoo.

You can access **odoo runbot** by going to `runbot.odoo.com/runbot`, as shown in the following screenshot:

The screenshot shows a web browser displaying the `runbot.odoo.com/runbot` page. The interface has a header with tabs for different Odoo editions: `odoorunbot`, `odoorunbot`, `odoodev/odoorunbot`, `odoodesign/themes`, `odoorunbot`, `odoodev/runbot`, and `Sign in`. Below the header is a search bar with fields for `Filter`, `Search`, and another `Search` button. A navigation bar at the top includes icons for back, forward, home, and refresh.

The main content area is a grid of cards representing different Odoo branches and their build statuses. Each card includes a title, a brief description, a progress bar, and several small icons for interacting with the build.

Branch	Build Status	Description	Progress (%)	Actions
10.0 1h Branch or pull H cov: 65.0%	[FIX] point_of_sale: lost pos or... Denis Ledoux 497720-10-0-39fd19 on runbot16.odoo.com ↑ age 1h time 25m	(coverage)[FIX] point_of_sale: l... Denis Ledoux 497556-10-0-39fd19 on runbot13.odoo.com ↑ age 3h time 45m	[FIX] point_of_sale: lost pos or... Denis Ledoux 497168-10-0-39fd19 on runbot20.odoo.com ↑ age 13h time 26m	[FIX] I10n_ca: add missing child... Olivier Colson 496958-10-0-e0ba54 on runbot11.odoo.com ↑ age 16h time 25m
11.0 55m Branch or pull H cov: 70.0%	[FIX] website_sale: default mail... Nicolas Martinelli 497726-11-0-309f15 on runbot21.odoo.com ↑ age 53m time 33m	(coverage)[FIX] website_sale: de... Nicolas Martinelli 497562-11-0-309f15 on runbot11.odoo.com ↑ age 3h time 51m	[FIX] website_sale: default mail... Nicolas Martinelli 497352-11-0-309f15 on runbot18.odoo.com ↑ age 10h time 33m	[IFIX] account: I10n_mx_edit inst... Johan Demaret Rivarola 497131-11-0-450101 on runbot15.odoo.com ↑ age 14h time 33m
12.0 53m Branch or pull H cov: 73.0%	[FIX] stock_account: price unit... Nicolas Martinelli 497730-12-0-e402c8 on runbot19.odoo.com ↑ age 53m time 41m	(click all)[FIX] stock_account: ... Nicolas Martinelli 497716-12-0-e402c8 on runbot15.odoo.com ↑ age 1h time 45m	[multi build][FIX] stock_account... Nicolas Martinelli 497615-12-0-e402c8 on runbot18.odoo.com ↑ age 1h time 40m	[multi build][FIX] stock_account... Nicolas Martinelli 497614-12-0-e402c8 on runbot11.odoo.com ↑ age 2h time 39m
master 1h Branch or pull H cov: 74.0%	[FIX] stock: Fix typo on field's... Swapnesh Shah (GitHub) 497718-master-f5c6c6 on runbot13.odoo.com ↑ age 1h time 50m	(click all)[FIX] stock: Fix tyo... Swapnesh Shah (GitHub) 497714-master-f5c6c6 on runbot22.odoo.com ⚠️ age 1h time 49m	[multi build][FIX] stock: Fix ty... Swapnesh Shah (GitHub) 497587-master-f5c6c6 on runbot14.odoo.com ↑ age 2h time 48m	[multi build][FIX] stock: Fix ty... Swapnesh Shah (GitHub) 497586-master-f5c6c6 on runbot14.odoo.com ⚠️ age 2h time 48m
saas-11.3 54m Branch or pull H cov: 74.0%	[MERGE] forward port branch 11.0... Christophe Simonis 497728-saas-11-3-372cef on runbot21.odoo.com ↑ age 54m time 31m	(coverage)[MERGE] forward port b... Christophe Simonis 497564-saas-11-3-372cef on runbot11.odoo.com ↑ age 3h time 49m	[MERGE] forward port branch 11.0... Christophe Simonis 496816-saas-11-3-372cef on runbot14.odoo.com	(coverage) [MERGE] forward port b... Christophe Simonis 496650-saas-11-3-372cef on runbot20.odoo.com

When you first navigate to **odoo runbot**, you are presented with a list of the Odoo Community builds. You will notice that there are branches for **10.0**, **11.0**, **12.0**, and the master branch, which is the current build that will eventually become Odoo 13. If you keep scrolling down, you will find many alternative builds that are for specific configurations or testing. For the purposes of this book, we will be focusing on the **12.0** branch.

Launching the Odoo Community Edition from Odoo runbot

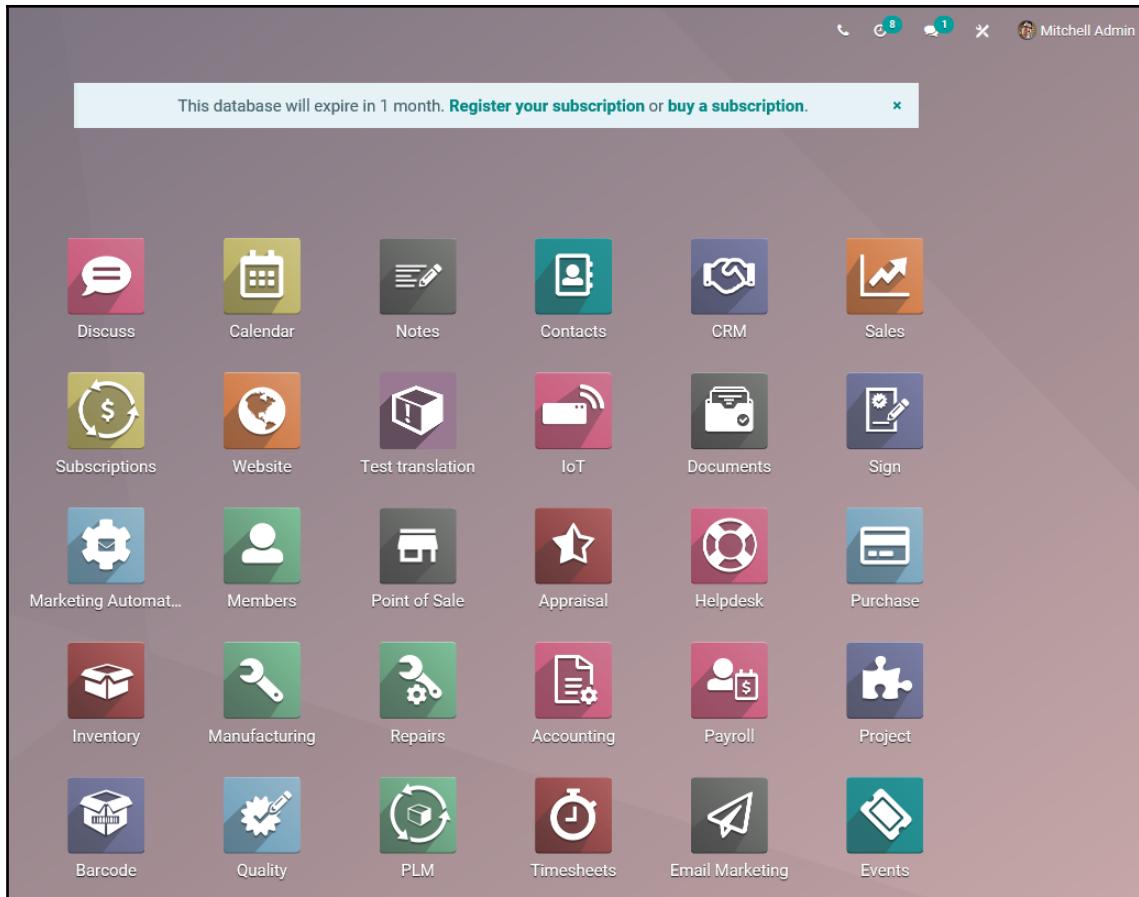
By using **odoo runbot**, we can launch any of the Odoo instances by clicking on the following small icon on the far left:



After you click the button, a new window will open, allowing you to choose between two databases.

One of the databases is just the base Odoo installation, while the other has all the Odoo applications. For our purposes, we are going to use the database that has all the applications already installed.

For all runbot installations, the email and password are `admin`. On validating the credentials we see the following screen:



After you have selected the database and entered the login information, you log in to the runbot instance just like you would in any version of Odoo.

Remember, this is the Community Edition of Odoo 12. While it is very probable that we will spend most of our time discussing Odoo Enterprise and the features that make it unique, it is still valuable to have the Community Edition open so as to make it easier to see the differences between the two.

Now, we will go back to the **odoo runbot** page and open the Enterprise Edition of Odoo in a separate window.

Launching Odoo 12 Enterprise from Odoo runbot

Now that you have Odoo 12 Community up and running in a window, let's proceed and start up Odoo 12 Enterprise in a separate window.

First, begin by opening a new browser window, or at least a new tab in the same browser. Then, simply navigate to **odoo runbot**, just like we did earlier in the chapter. This time, however, instead of launching the Community branch, use the link at the very top, labeled **odoo/enterprise**.

After you click the link, the runbot page will refresh and show the branches for the Enterprise version of Odoo.

Now, just click on the **connect** button and log in to the Odoo Enterprise Edition the same way you did the Community version.

You should now have both Odoo Community and Odoo 12 Enterprise running in separate windows on your computer, making it easy to switch back and forth in order to compare features. Best of all, you didn't have to provide an email, install any software, or pay any license fees!

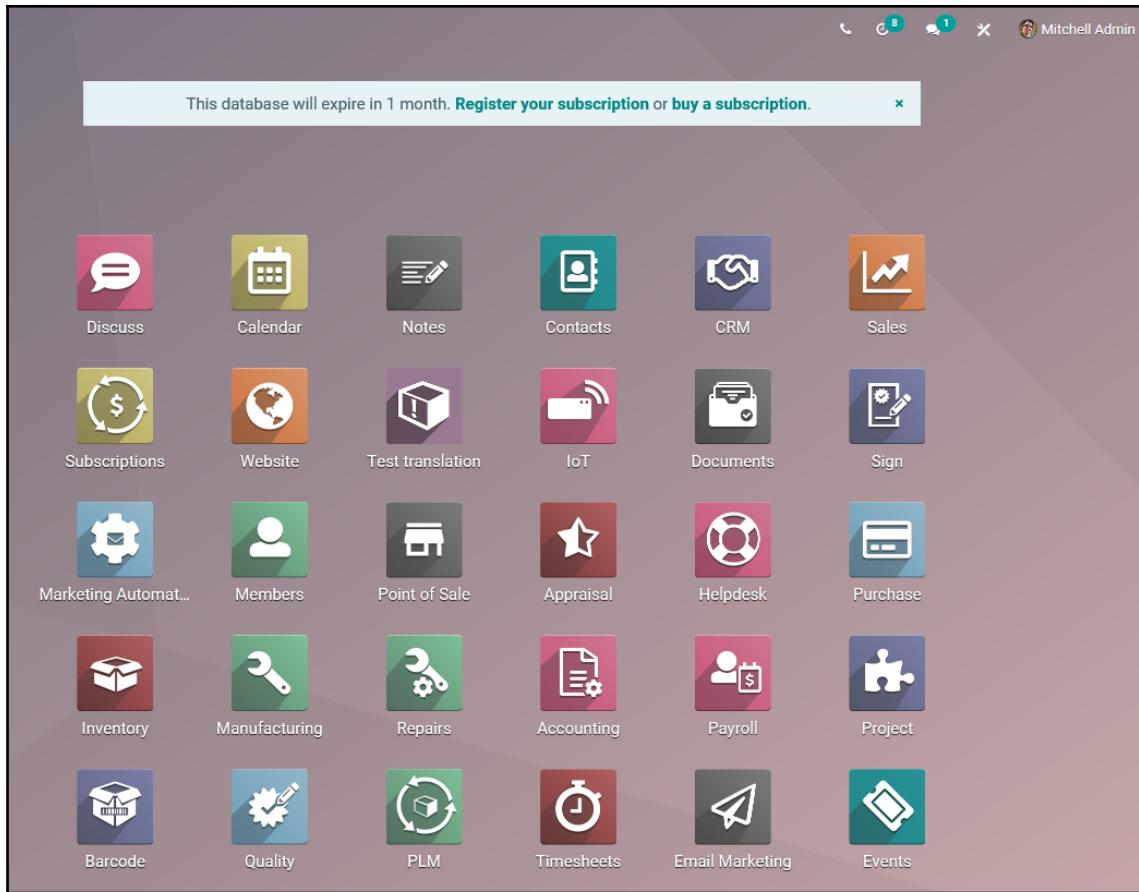


WARNING: Unlike a standard Odoo installation, runbot installations are only temporary, and anything you put into them can be wiped out at any time. The runbot is great for situations like this, when you just need to check some specific Odoo functionality.

Finally, take the time to arrange the windows in a way that works best for your current workspace and monitor configuration. For example, if you have two monitors, you could put the Community Edition on one monitor and the Enterprise Edition on the other monitor so that you easily compare the functionality of the two versions.

Examining the Odoo Enterprise interface

Without a doubt, the most noticeable difference between Odoo Community and Odoo Enterprise is the interface. In fact, the entire set of application menus has disappeared and been replaced with a page of icons that allow you to launch any of the installed applications:



As we are only using this Odoo 12 Enterprise installation to check out the features, you can ignore the database expiration notice at the top. Also, the exact applications that are available may be slightly different, or in different locations than the previous example. Remember that other people are accessing these same instances; therefore, you can't be certain that settings will be exactly the same as a default installation.

Examining Odoo 12 Enterprise Accounting

While the interface enhancements are easily the most obvious differences between Odoo Community and Odoo Enterprise, the reports in the **Accounting** application are perhaps the most valuable functional improvements over their Community version equivalents.

Let's look at the dynamic Enterprise reports by clicking on the **Accounting** icon in order to launch the **Accounting** application:



When the application opens, you will see a dashboard that includes your invoices, vendor bills, and the critical journals of your company. The **Accounting** application also includes the new application configuration wizard that has been introduced in Odoo 12. As you can see here, the wizard steps you through setting up your company information, bank account information, preferences for fiscal year, and your accounts:

A screenshot of the Odoo Accounting application interface. At the top, there is a navigation bar with tabs: Overview, Customers, Vendors, Accounting, Reporting, and Configuration. On the right side of the header, there are user-related icons and a dropdown for 'Mitchell Admin'. The main area is titled 'Accounting Overview' and features a 'Company Data' configuration section with four buttons: 'Let's start!', 'Add a bank', 'Configure', and 'Review'. Below this, there are four main report cards: 'Customer Invoices' (Sale), 'Vendor Bills' (Purchase), 'Bank', and 'POS Sale Journal' (Sale). Each card provides summary data and buttons for 'NEW INVOICE', 'NEW BILL', 'RECONCILE 5 ITEMS', and 'CONFIGURE'. The 'Customer Invoices' card shows 0 Invoices to Validate and 4 Unpaid Invoices totaling \$10,495.00. The 'Vendor Bills' card shows 2 Bills to Validate, 0 Bills to Pay, and 0 Expenses to Process. The 'Bank' card shows a balance of \$102.78 and a difference of \$9,745.42. The 'POS Sale Journal' card shows 0 Invoices to Validate and 0 Unpaid Invoices.

One of the primary differences we will examine in this chapter is the accounting reports that come with Odoo Enterprise.

Looking at Odoo 12's dynamic accounting reports

In previous versions of the Community Edition of Odoo, you still had access to PDF versions of the accounting reports. In Odoo 12 Community, the accounting reports have been removed entirely.



You can still get the PDF versions of the accounting reports in Odoo 12 Community by using relatively inexpensive options available in the Odoo Apps.

Let's look at the Odoo Enterprise Profit and Loss statement by choosing **Profit and Loss** from the **Reports** menu:

A screenshot of the Odoo Accounting module showing the Profit and Loss report. The top navigation bar includes links for Overview, Customers, Vendors, Accounting, Reporting, Configuration, and a user dropdown for 'Mitchell Admin'. Below the navigation is a toolbar with 'PRINT PREVIEW' and 'EXPORT (XLSX)' buttons, and filters for '2019', 'Comparison: No comparison', 'Journals: All', 'Analytic', and 'Options: Accrual Basis, Posted Entries Only'. The main content area displays the 'Profit and Loss' report for 'YourCompany' for the year 2019. The report is organized into sections for Income, Expenses, and Net Profit, with detailed breakdowns for each category. The data is presented in a table format with dollar amounts.

If you take the time to switch over to Odoo Community and attempt to produce the same **Profit and Loss** report in PDF format (which now requires you to install extra third-party applications), you can quickly appreciate the improvements made in Odoo Enterprise. Even better is the fact that the reports are dynamic, allowing users to drill up, drill down, and filter data on the fly. There are a variety of options at the top to change the filters used on the report and to perform comparisons between different periods. For instance, as shown with this **Profit and Loss** report, you can filter by a variety of preset date ranges or define custom filters:

The screenshot shows the Odoo Accounting module's Profit and Loss report. At the top, there are buttons for 'PRINT PREVIEW' and 'EXPORT (XLSX)'. To the right, there are filter options: a date range selector set to '2019' with dropdowns for 'Comparison: No comparison', 'Journals: All', and 'Accrual Basis, Posted Entries Only'; and a dropdown menu for financial periods: 'This Month', 'This Quarter', 'This Financial Year' (which is checked), 'Last Month', 'Last Quarter', 'Last Financial Year', and 'Custom'. The main area displays a table of income and expenses. The 'Income' section shows 'Gross Profit' (\$10,597.78), 'Operating Income' (\$10,597.78), 'Cost of Revenue' (\$0.00), and 'Other Income' (\$0.00). The 'Expenses' section shows 'Expenses' (\$14,999.99) and 'Depreciation' (\$0.00). The total profit is \$-4,402.21.

Category	Amount
Income	\$ 10,597.78
Gross Profit	\$ 10,597.78
Operating Income	\$ 10,597.78
Cost of Revenue	\$ 0.00
Other Income	\$ 0.00
Expenses	\$ 14,999.99
Expenses	\$ 14,999.99
Depreciation	\$ 0.00
Total Profit	\$ -4,402.21

Another nice feature is the fact that the reports have drill-down capabilities. For example, you can open **Operating Income** and click on the **Product Sales** account, which brings up a menu that will take you to the general ledger for the account, or allow you to add a note to the item that is on the report:

The screenshot shows the NetSuite Accounting module interface. At the top, there's a navigation bar with 'Accounting' selected, along with links for Overview, Customers, Vendors, and various notifications. On the left, a sidebar displays the 'Profit and Loss' report for 'YourCompany'. The main area shows a table with two sections: 'Income' and 'Expenses'. The 'Income' section includes categories for Gross Profit, Operating Income (which is expanded), Cost of Revenue, and Other Income. The 'Expenses' section includes categories for Expenses and Depreciation. A context menu is open over the 'Operating Income' row, listing options like 'This Month', 'This Quarter', and 'This Financial Year'. The 'This Financial Year' option is currently selected. The table data is as follows:

	2019
This Month	
This Quarter	
This Financial Year	\$ 10,597.78
Last Month	
Last Quarter	
Last Financial Year	
Custom	

	2019
Gross Profit	\$ 10,597.78
Operating Income	\$ 10,597.78
Cost of Revenue	\$ 10,597.78
Other Income	\$ 0.00
Expenses	\$ 14,999.99
Expenses	\$ 14,999.99
Depreciation	\$ 0.00

Even though the reports show up immediately, this does not mean, however, that you give up the option to have them in PDF format. To the top-left side of the reports, you have buttons that allow you to export the report in PDF or Excel formats.

Finding out which options are only available for Enterprise

While dynamic reports may be the flashiest part of the Enterprise **Accounting** application, the Enterprise version also offers more robust import options. To get a better idea of what options are available in the Enterprise Edition, you can go to the **Settings** menu and then choose **General Settings** in the Community Edition. Simply click on an application and you will see a small **Enterprise** tag next to the options that are only available in the Enterprise Edition of Odoo. In the following example, we have selected the **Manufacturing** application and can see the options that require the Enterprise edition:

The screenshot shows the Odoo Settings interface for the Manufacturing application. On the left, a sidebar lists various applications: CRM, Sales, Website, Purchase, Inventory, Manufacturing (selected), Invoicing, Payroll, Project, Timesheets, Email Marketing, Events, Employees, Recruitment, Attendances, Expenses, Point of Sale, and General Settings. The main content area is titled 'Operations' and contains three sections: 'Work Orders' (with sub-options 'Routings' and 'Work Centers'), 'By-Products' (described as 'Produce residual products (A + B -> C + D)'), and 'Quality Enterprise' (described as 'Add quality checks to your work orders'). Below this is a 'Products' section with 'By-Products'. The 'Planning' section includes 'Master Production Schedule Enterprise' (described as 'Plan manufacturing or purchase orders based on forecasts') and 'Security Lead Time' (described as 'Schedule manufacturing earlier to avoid delays'). At the top of the main content area, there are 'Save' and 'Discard' buttons, and a search bar.

It is far beyond the scope of this chapter to cover each of these options. You will notice that at the very top is an option to view **Work Orders** and **Quality**. We will look at these Enterprise applications later in the chapter. Further down the screen, under **Planning**, you will see that there is an option for a **Master Production Schedule**. Fortunately, from what you learned earlier, you can use the runbot to easily see what the various features look like in the Enterprise Edition.

Depending on your own requirements, you may find certain Enterprise options very important in integrating Odoo into your business.

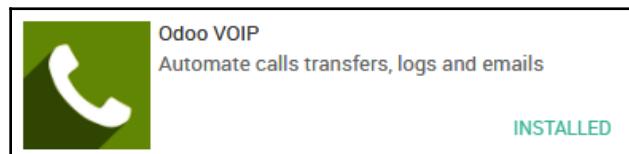
Sales-related enhancements

As **Sales** is a very popular Odoo application, it is not surprising that the Enterprise Edition of Odoo contains several application that are not in the Community Edition, but that are designed to provide additional features. While not as significant as the dynamic reports or enhanced interface, the Enterprise options in **Sales** can be quite useful, depending on your business requirements.

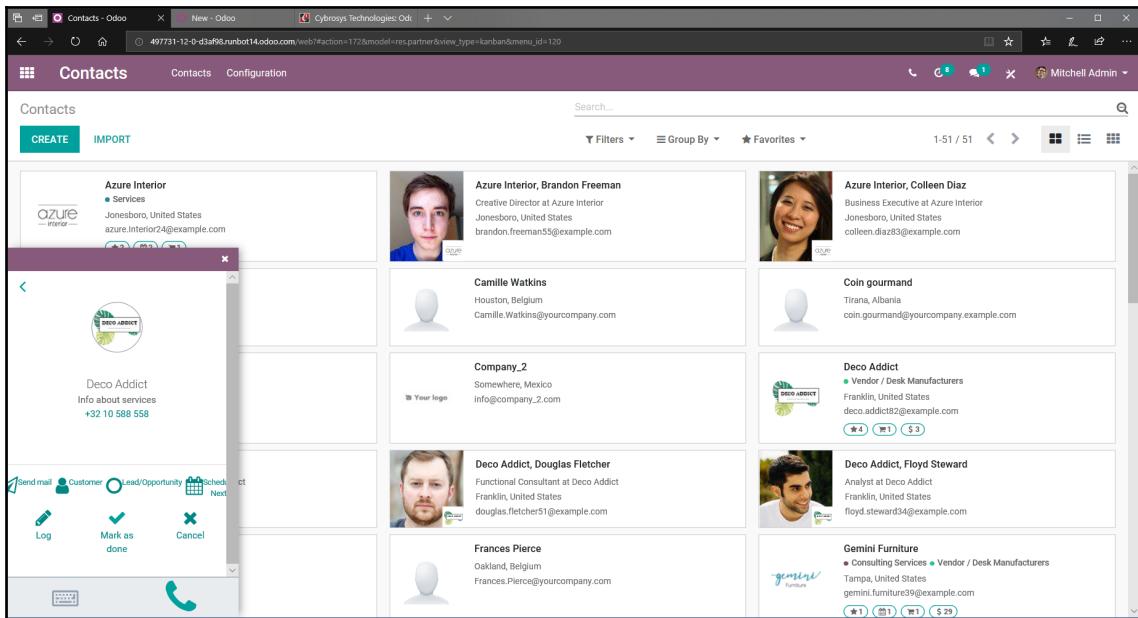
Automating calls with VoIP integration

If you have a sales team that has to make or take calls on a frequent basis, the built-in VoIP integration in Odoo Enterprise can be a real time-saver. Instead of your sales team having to look up incoming calls within Odoo manually, Odoo's VoIP integration will automatically locate the contact within Odoo when the call comes in.

Unlike other Enterprise options, if you are installing VoIP in an Enterprise installation yourself, you will need to add the **Odoo VOIP** application:



Using the VoIP module to play calls is easy. You can bring up the dialing panel by clicking on the small phone icon in the top-center of the screen. You also can click on any phone number field in Odoo and bring up the dialing panel. All the calls that you need to place will be listed, along with several other buttons that let you handle how you manage the call. The following screenshot shows you the list of contacts with the dialing panel activated:



Naturally, to configure and set up the **VoIP** application, you will require some technical expertise to be able to integrate it properly with your phone system.

Understanding the Subscriptions application

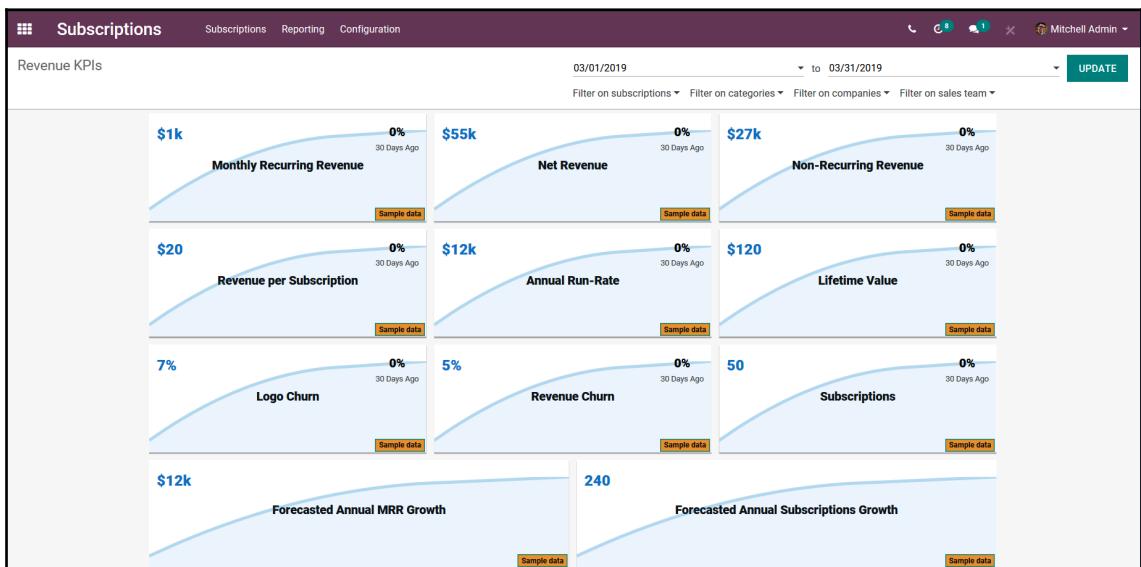
Like the VoIP module, the **Subscriptions** application is also installed separately.

When you open the **Subscriptions** application, you are presented with a kanban view that provides you with an overview of your current subscriptions as shown in the following screenshot:

The screenshot shows a kanban board titled "Subscriptions". It has three columns: "Draft" (0 items), "In Progress" (3 items), and "To Upsell" (2 items). Each item in the lists has a star icon, a circular progress bar, and a small "To Renew" button.

Column	Subscription Details
Draft	YourCompany, Joel Willis SUB001 \$ 20.00 To Renew
In Progress	YourCompany, Joel Willis SUB008 \$ 20.00 To Renew
In Progress	YourCompany, Joel Willis SUB009 \$ 20.00 To Renew
To Upsell	YourCompany, Joel Willis SUB006 \$ 20.00 To Renew
To Upsell	YourCompany, Joel Willis SUB007 \$ 15.00 To Renew

Additionally, you can also get a full-featured **key performance indicators (KPIs)** dashboard that lets you track your subscriptions. You can see this by choosing **Revenue KPIs** from the **Reporting** menu shown as follows:



Subscriptions are configured as templates that allow you to easily see critical information such as the number of subscribers. Choose **Subscription Products** under the **Subscriptions** menu to see the current templates that are set up in the system. The following screen appears after clicking:

The screenshot shows the 'Subscriptions' application interface. At the top, there are tabs for 'Subscriptions', 'Reporting', and 'Configuration'. On the left, there are 'CREATE' and 'IMPORT' buttons. The main area displays a list of 'Subscription Products' with the following columns: Internal Reference, Name, Website, Sales Price, Cost, Product Category, Product Type, Quantity On Hand, Forecasted Quantity, and eBay Status. Two entries are listed:

Internal Reference	Name	Website	Sales Price	Cost	Product Category	Product Type	Quantity On Hand	Forecasted Quantity	eBay Status
SERV_89553	Office Cleaning Subscription (Monthly)		200.00	0.00	All / Saleable / Services	Service			Unlisted
SERV_89137	Office Cleaning Subscription (Yearly)		2,000.00	0.00	All / Saleable / Services	Service			Unlisted

As you can see, the subscription template displays important information about the subscriptions in a list format by default. Clicking on a subscription template will display the details of the subscription. As you can see in the following screenshot, there are many options:

The screenshot shows the detailed view of a subscription product. At the top, it says 'Subscription Products / [SERV_89553] Office Cleaning Subscription (Monthly)'. There are 'EDIT' and 'CREATE' buttons, as well as 'Print' and 'Action' dropdowns. The main area shows the product details with various status indicators:

- Unpublished On Website
- 0 Attachments
- 0.00 Month(s) Purchased
- 0.00 Month(s) Sold
- 0 Digital Files
- Active

The product name is 'Office Cleaning Subscription (Monthly)'. Below it, there are several checkboxes for configuration:

- Can be Sold
- Can be Purchased
- Can be Expensed
- Is a Landed Cost
- Sell on eBay

The product type is 'Service', category is 'All / Saleable / Services', internal reference is 'SERV_89553', sales price is '\$ 200.00', customer taxes are 'Tax 15.00%', and cost is '\$ 0.00'. The version is '1'. There is also a section for 'Internal Notes'.

Like many of the applications in Odoo, it is important to take the time to build your experience of configuring subscriptions and testing different scenarios to find out what works for you. If you are running a business that has subscription-based sales, the **Subscriptions** application is a strong reason to consider Odoo Enterprise.

Understanding the available shipping connectors in Enterprise

As many companies are involved with shipping products, the availability of shipping connectors in Odoo Enterprise could be a major consideration for some companies. These Enterprise shipping connectors hook into Odoo Inventory and eCommerce applications, allowing you to automatically create the required transactions inside the selected shipping system.



If you only need Odoo Enterprise for shipping connectors, or another feature or two, check the Odoo App Store. There are many third-party Odoo applications that work with Odoo Community and offer similar functionality to Odoo Enterprise.

To configure your Odoo shipping connectors, go into the **Inventory** application, then the **Configuration** menu, and then click **Settings**:

The screenshot shows the Odoo Inventory application interface. The top navigation bar includes 'Overview', 'Operations', 'Master Data', 'Reporting', and 'Configuration'. On the far right, there are notification icons for messages, calls, and errors, along with a user profile for 'Mitchell Admin'. Below the navigation, a sidebar lists various Odoo modules: CRM, Sales, Website, Purchase, Inventory (which is currently selected), Manufacturing, Accounting, Payroll, Project, Timesheets, Email Marketing, Events, Employees, and Documents. The main content area is titled 'Settings' and contains a 'Shipping Connectors' section. This section lists six connectors, each with a checked checkbox and a brief description: 'UPS Connector' (Compute shipping costs and ship with UPS), 'FedEx Connector' (Compute shipping costs and ship with FedEx), 'bpost Connector' (Compute shipping costs and ship with bpost), 'DHL Connector' (Compute shipping costs and ship with DHL), 'USPS Connector' (Compute shipping costs and ship with USPS), and 'Easypost Connector' (Compute shipping costs and ship with Easypost). Each connector entry includes a link to 'Delivery Methods'. Below the shipping connectors, there is a 'Products' section with two items: 'Variants and Options' (Set product attributes (e.g. color, size) to manage variants) and 'Units of Measure' (Sell and purchase products in different units of measure). Each product item has a corresponding checkbox and a link to its respective documentation.

After you have checked which shipping connectors you would like to be available as shown in the preceding screenshot, you can use the **Delivery Methods** option under the **Configuration** menu to see all the connectors and configure them as required:

<input type="checkbox"/> Delivery Method	Provider	Visible on current website	Website
<input type="checkbox"/> Free delivery charges	Fixed Price	<input checked="" type="checkbox"/>	
<input type="checkbox"/> The Poste	Based on Rules	<input type="checkbox"/>	
<input type="checkbox"/> Normal Delivery Charges	Fixed Price	<input type="checkbox"/>	
<input type="checkbox"/> Bpost World Express Pro	bpost	<input type="checkbox"/>	
<input type="checkbox"/> Bpost Domestic bpack 24h Pro	bpost	<input type="checkbox"/>	
<input type="checkbox"/> DHL USA	DHL	<input type="checkbox"/>	
<input type="checkbox"/> DHL USA -> International	DHL	<input type="checkbox"/>	
<input type="checkbox"/> FedEx International	FedEx	<input type="checkbox"/>	
<input type="checkbox"/> FedEx US	FedEx	<input type="checkbox"/>	
<input type="checkbox"/> UPS US	UPS	<input type="checkbox"/>	
<input type="checkbox"/> UPS BE	UPS	<input type="checkbox"/>	
<input type="checkbox"/> USPS Domestic Flat Rate Envelope	USPS	<input type="checkbox"/>	
<input type="checkbox"/> USPS International Flat Rate Box	USPS	<input type="checkbox"/>	

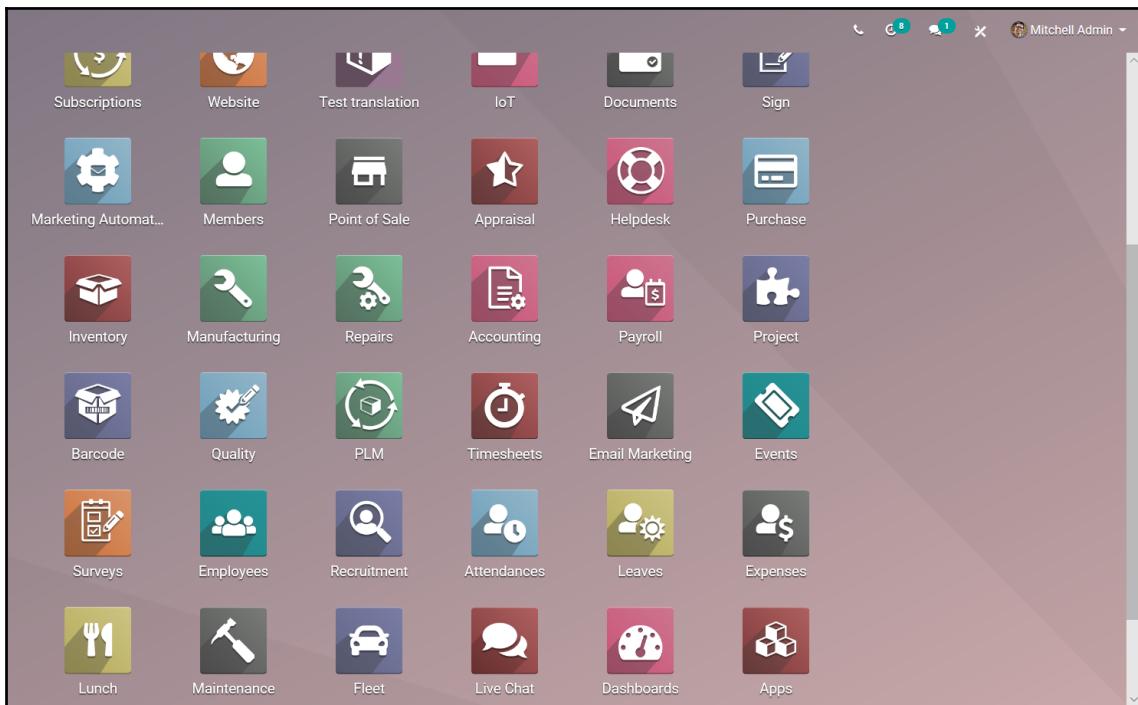
As you can see from the preceding list, Odoo Enterprise offers quite a few connectors. As with VoIP integration, you will require some technical expertise and all the configuration information from your shipping system in order to configure the connector properly.

Understanding the available manufacturing applications in Odoo

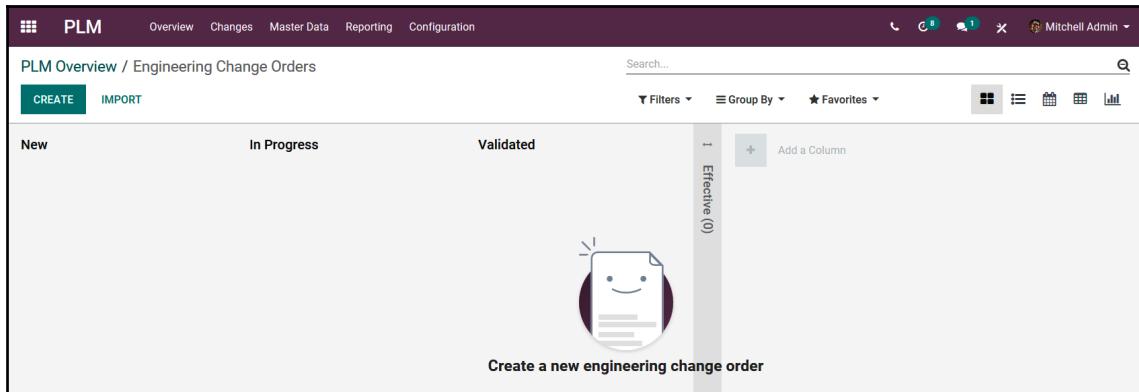
Perhaps, of all the applications that the Enterprise Edition offers, the new manufacturing applications provide some of the strongest incentives for considering the Enterprise Edition, should they satisfy your business requirements. These applications include **PLM**, **Maintenance**, and **Quality**.

Product Lifecycle Management (PLM)

When you are working in a production environment, it is likely, if not inevitable, that there will be changes to the build process of various items. Perhaps you get a new vendor for a part, or a small design change requires you to add or remove parts from a bill of material. Managing these changes can be very time-consuming. This is where the PLM application comes into play. You can access it by clicking on the **PLM** application icon from the main Enterprise menu:

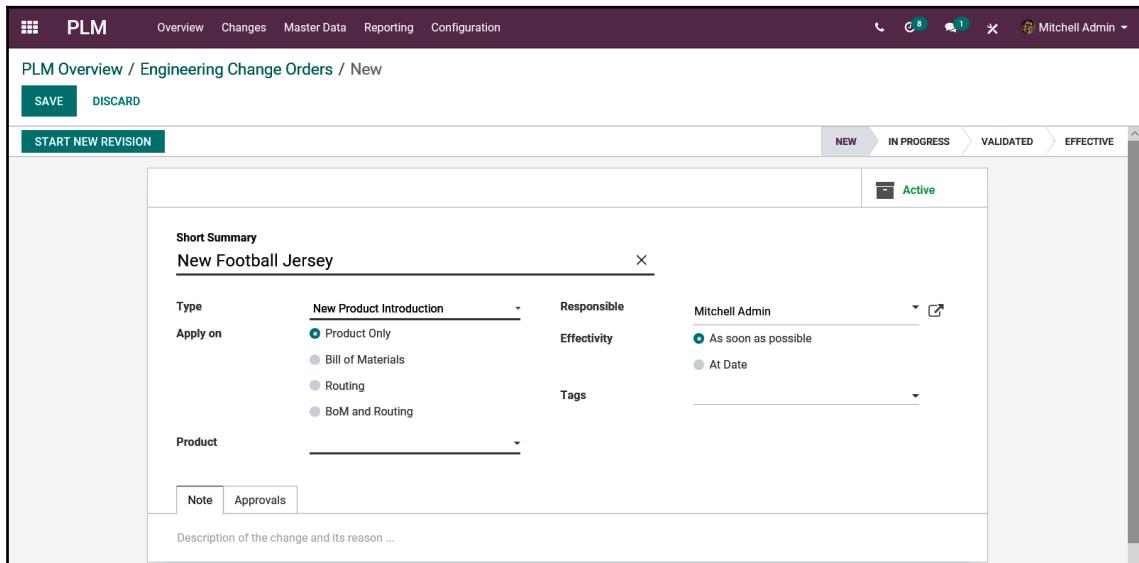


Clicking on the **PLM** application icon will take you to the **PLM** dashboard, which, in this demo version, is lacking any real data. Click on the **Engineering Changes** button to bring up the kanban view, which displays the stages that are currently defined for the product life cycle:



As shown in the preceding screenshot, the following stages, **New**, **In Progress**, and **Validated**, are all expanded. The **Effective** stage is used when the cycle is complete.

Now, let's see what an **Engineering Change Order** looks like in the **PLM** application by clicking on the **Create** button:



We have filled in a little bit of information on the form so that you can see an example of how it may be used. You will notice that we have a **Short Summary** form that will describe the change that must be managed.

The **Type** defaults to **New Product Information**. The PLM application is designed to work for a new product whose cycle you wish to manage; alternatively, you can create a change order for an existing product.

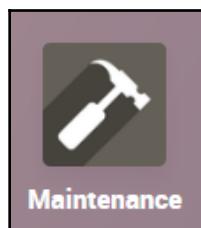
One nice thing about the PLM application is that, in addition to managing changes on your **Bill of Materials (BoM)**, it will also let you manage changes on the routing, or on both the BoM and the routing. This gives you a great deal of flexibility when managing changes in the production process.

Naturally, to complete our change order, we must know what **Product** the order is for and the associated **Bill of Materials**. You can add each of these fields on the fly, making it easy to configure a new product and the associated change order from this very same **Short Summary** screen.

Finally, the PLM application provides an effective date so that you can decide exactly when you want the change order to go into effect. This makes it easy to manage complex changes that are forthcoming in future product builds. One example would be a model year change-over, in which you want everything on your BoM to change on a specific date, but this could take many weeks of configuration ahead of time.

Handling maintenance requests

Another common requirement in manufacturing operations is the handling of maintenance requests. This can be a machine that is malfunctioning, or even a work environment that needs attention to be more productive. The Odoo **Maintenance** application integrates the handling of maintenance requests directly into the manufacturing process. Click the **Maintenance** icon on the main **Enterprise** menu to bring up the **Maintenance** dashboard:



You will then be taken to the **Maintenance Teams** dashboard:

Maintenance Team	Scheduled	Top Priorities	Blocked	Unscheduled
Internal Maintenance	0	0	0	3
Metrology	0	0	0	0
Subcontractor	0	0	0	0

You will see that the Odoo **Maintenance** application divides our maintenance requests by teams. In this case, we have three teams configured to handle the requests. Under the **Internal Maintenance** team, we can click the button that says **3 TO DO** and pull up the requests shown as follows:

Status	Description	Category	Priority
New Request	Some keys are not working	Acer Laptop/LP/203/19281928	Computers
In Progress	Touchpad not working	HP Laptop/LP/305/17281718	Computers
In Progress	Battery drains fast	HP Laptop/LP/303/28292090	Computers
Repaired (0)			

Here, we can see that one of the requests is new, while the other two requests are in progress. Click on the **Battery drains fast** maintenance request to view the details:

Battery drains fast

Created By	Team
Equipment	Internal Maintenance
Category	Marc Demo
Request Date	Scheduled Date
Maintenance Type	Duration
Manufacturing Order	Priority

Send message Log note Schedule activity

OdooBot · 2 hours ago Status changed

- Request Date: April 11, 2019
- Category: Computers
- Technician: Marc Demo

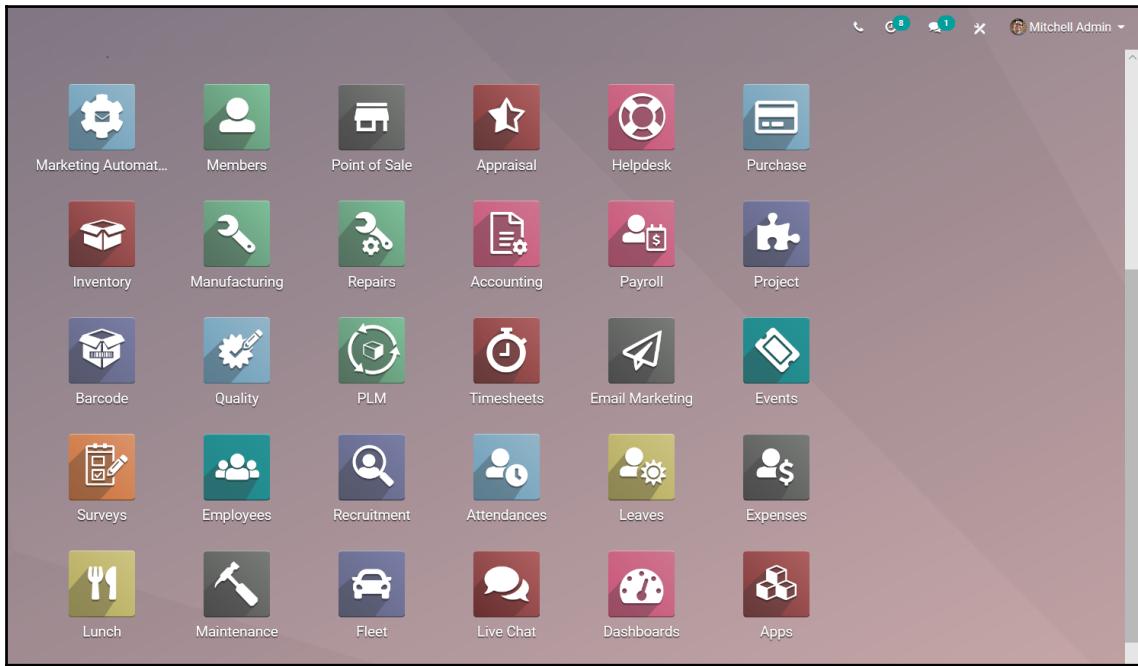
As you can see, there are a lot of details on this request that help us to identify exactly what needs to be done in order to address the problem. In our example, we know who the request was **Created By**, what **Equipment** was involved, and the **Request Date**, among other important items.

Additionally, Odoo will automatically fill in the **Manufacturing Order** or the **Work Order** if a maintenance request is made during one of those processes. When you are ready, you can set a **Scheduled date** for the repair, as well as a priority level to help you manage what items your maintenance team should be focused on.

Managing production quality in Odoo

As your team is manufacturing items, it is inevitable that there will be situations in which you will need to address the quality of the products. This could be because you received raw materials that were not adequate; it could be because there is a problem in the process; or it could be something that requires further investigation to determine why a product is not meeting your quality expectations. Regardless of the reason Odoo's **Quality** application can help you to manage quality better in your manufacturing operations.

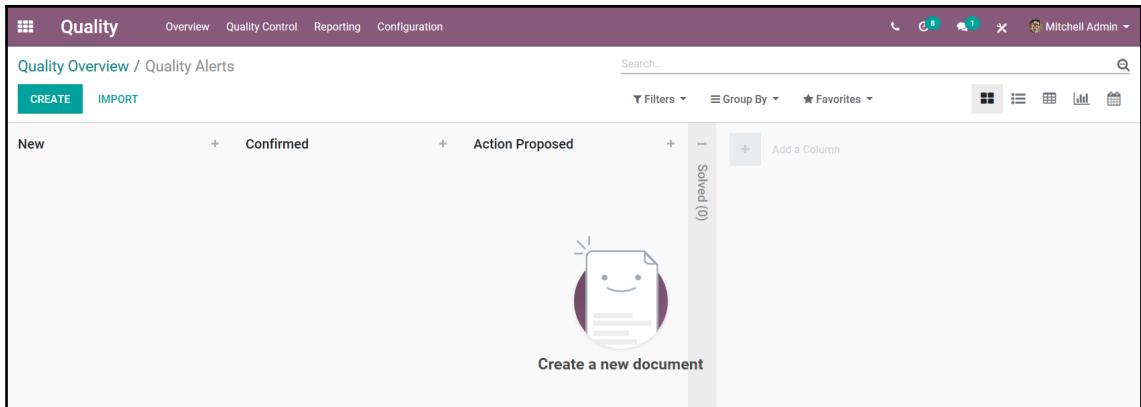
Click the **Quality** icon from the main **Enterprise** menu to view the **Quality** dashboard:



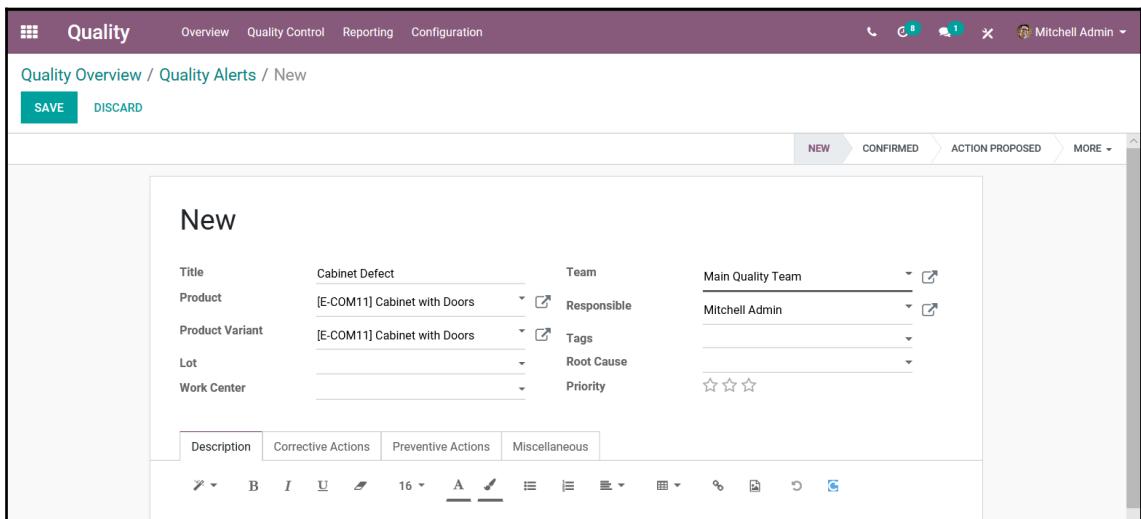
Like the **Maintenance** application, the **Quality** application breaks quality alerts down into different teams. On the dashboard, you can see that there is currently one quality team defined; the **Main Quality Team**:

A screenshot of the Quality dashboard. The top navigation bar includes icons for Quality, Dashboard, Quality Control, Reporting, and Configuration. The main content area is titled "Quality Dashboard". Below it, a section is titled "Main Quality Team". Inside this section, there is a teal button labeled "0 QUALITY ALERTS". To the right of the button, it says "Checks In Progress" and shows the number "0". There is also a small three-dot menu icon in the top right corner of this section.

While we don't have any active **Quality Alerts**, let's proceed and click on the button so that we can see the stages that have been set up by default to manage our quality alerts: The following screenshot depicts an empty overview of the quality alerts:



As you can see, **PLM**, **Maintenance**, and **Quality** are all designed in a similar way. As always, we can click **Create** to enter a quality alert:



Here, you can describe the product, the variants, and the lot the product was in, given that the quality issue is associated with a specific lot. Items that you specify in **Corrective Actions** are automatically presented when someone begins a manufacturing order. This makes it easy to quickly alert someone in manufacturing when there is a quality issue, and provide them with clear instructions on what they need to do to verify or, if necessary, fix the product quality.

Using Odoo Studio

Odoo Studio is an Odoo Enterprise-only application that allows you to customize Odoo and even create your own custom applications without creating a Python module, as discussed previously in Chapter 15, *Discovering Custom Odoo Modules*. There are some pros and cons to using Odoo Studio.

The advantages of using Odoo Studio are as follows:

- It allows simple changes to Odoo applications without writing Python code.
- Screen designers make it easy to change the UI without verbose XML code.
- It prototypes applications in a fraction of the time it would take to create a full Odoo application.
- Using server actions, you can get close to full-blown Odoo applications.
- You can import to and export from Odoo Studio, giving you the ability to reuse your work between Odoo Enterprise installations.
- Modifying the views in Odoo Studio and the capabilities that are available to you are pretty much the same as if you were developing an Odoo application.

The disadvantages of using Odoo Studio are as follows:

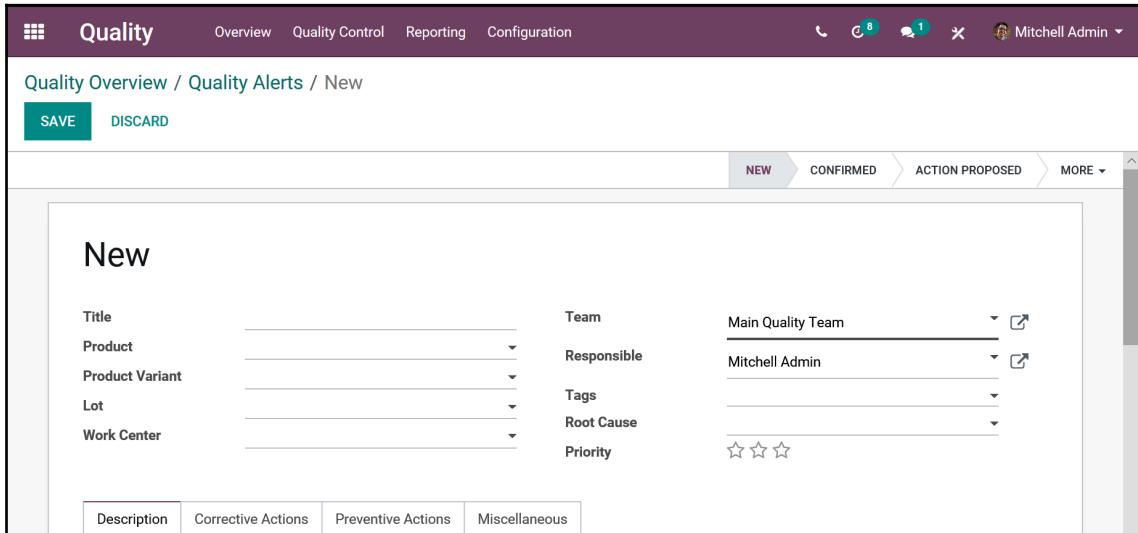
- Nothing you do in Odoo Studio can be used in the Community Edition of Odoo.
- Despite many enhancements to Odoo 12, there are still major limitations compared to writing custom Odoo applications.
- Because of the way you develop with Odoo Studio, this makes it very difficult to manage your version control within production installations.

Modifying the Sales application with Odoo Studio

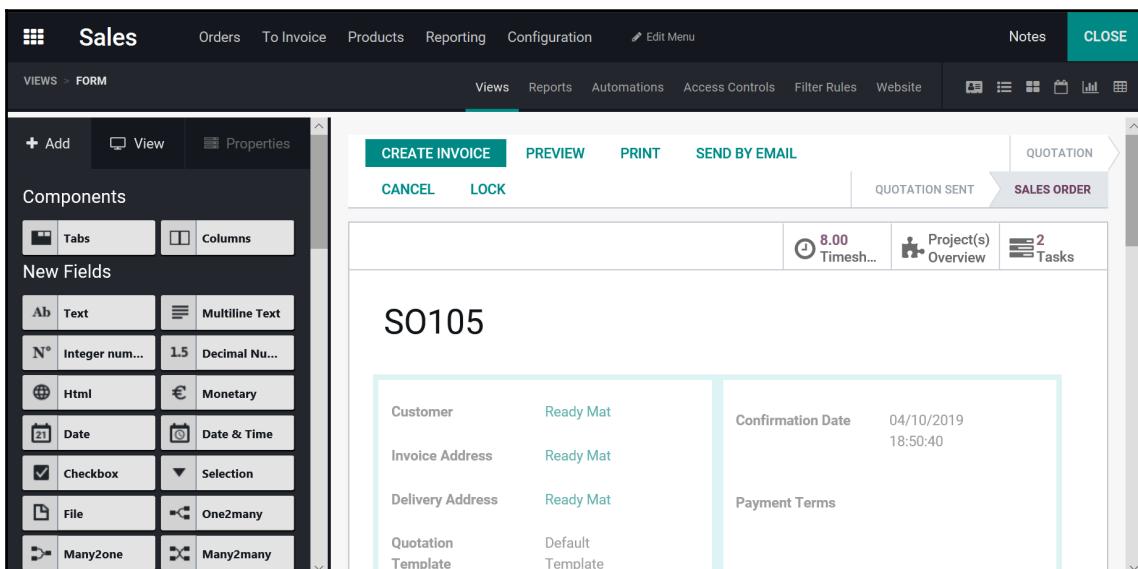
You can install Odoo Studio as you would any other application. Remember, however, that you must have Odoo Enterprise to use Odoo Studio. Once again, we can use the runbot to try out some of Odoo Studio's features.

As Odoo Studio is already installed on the runbot, we can simply go to the application we wish to modify to get started. Click on the **Sales** menu and open a **Quotation**.

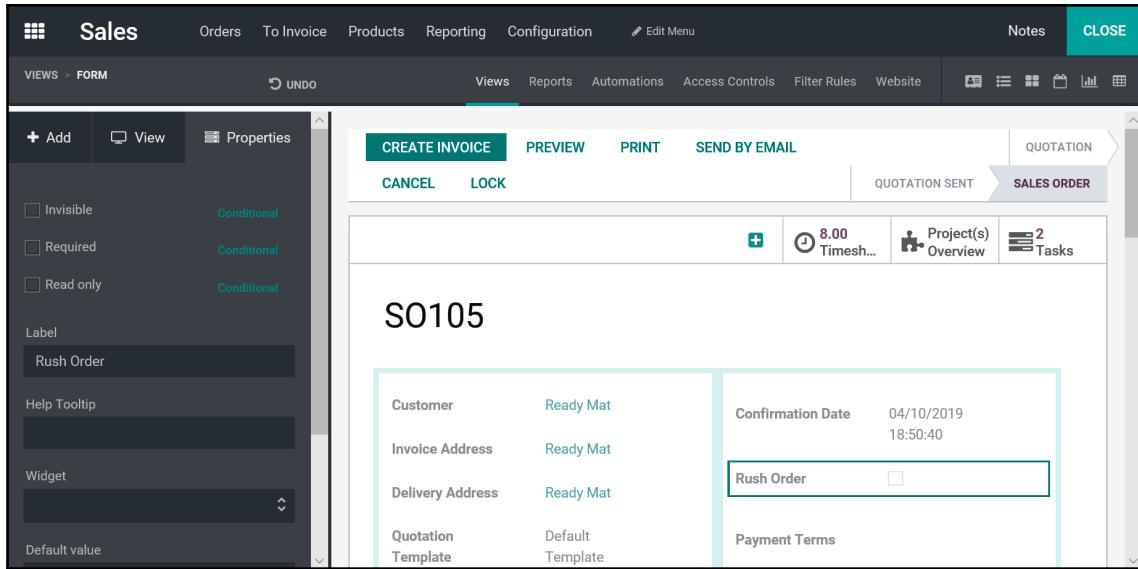
Now, to edit the **Quotation**, you can click on the **Odoo Studio** edit button in the upper right corner of the interface. It looks like a crossed wrench and pencil:



Once you turn on **Odoo Studio**, the interface will change to show you tools that you can use to modify the form. So it is very plain to see that you are editing your application when you are in **Odoo Studio**. Note the toolbar along the left side of the screen:



You can just as easily close out of **Odoo Studio** by hitting the large green **CLOSE** button in the upper right corner. Now, let's add a **Rush Order** checkbox to our form. Simply click and drag the **Checkbox** from the right and place it on the form where you wish to display it. In the following example, we placed our **Rush Order** checkbox between **Confirmation Date** and **Payment Terms** as shown in the following screenshot:



At this point, you can edit the properties for the checkbox so that it will display as you wish. Take the time to experiment with the options available under **Properties**. When you click the **CLOSE** button, the form will refresh and you will see your changes applied.

This was clearly just a brief introduction, and it would be easy to dedicate a whole chapter to Odoo Studio, or perhaps even an entire book. There are a lot of options available in Odoo Studio, and you can expect to take some time to learn about how they all fit together.

Summary

In this chapter, we learned about some of the more important features that are part of Odoo Enterprise. We learned a bit about the new interface and how Odoo Enterprise restructures the menus to better use screen real estate. We took a look at the powerful dynamic accounting reports, as well as the bank account integration feature. Next, we examined the Sales application and studied how Odoo Enterprise offers VoIP integration as well as the Subscriptions application. We examined Odoo Enterprise's shipping connectors and how they could be used to connect with external shipping systems. Finally, we took a look at Odoo's new manufacturing applications that are only available in the Enterprise Edition.

In this book, we have covered a wide range of topics in an attempt to give you a solid foundation. We began by trying out the online version of Odoo and performing an installation of Odoo on Windows or Ubuntu. As we moved through the book, we learned about the major applications that make up the Odoo business platform, as well as some real-world examples of how they may be used. Later in the book, we learned how to administer and customize Odoo, as well as how to use Odoo as a content management system for your website. Finally, we finished by exploring the basics of customizing Odoo and even building our own Odoo applications.

Like any ERP system, becoming an expert in Odoo will require many hundreds of hours of research, experimentation, and hard work. With enough practice, I'm confident that you will be able to configure and customize Odoo as needed to suit the essential demands of your business and your clientele. Best of luck to you!

Locating Additional Odoo Resources

Odoo is built using a variety of open source technologies and components. This appendix includes a list of resources that can extend your knowledge when it comes to supporting an Odoo installation.

Locating the essential Odoo documentation

If you are looking for the official Odoo documentation, then the link for you is: <https://www.odoo.com/documentation>.

At the time of writing, this documentation automatically directs you to the latest Odoo version and provides direct access to the primary technical documentation. The quality and scope of the documentation has increased greatly over the past several years. This is where you will want to look first when you want to learn more about Odoo development.

Visiting the official Odoo help forum

When you run into issues with your Odoo installation or you have questions about specific features, one of the best resources that's available is the official Odoo help forum. The site is simple in that you can search for questions, and if your specific question hasn't been addressed, you can then ask it yourself. This has become a very active forum and is worth a visit if you need help with a problem. You can find it at <http://help.odoo.com>.

The Odoo Community Association

The **Odoo Community Association (OCA)** is a nonprofit organization whose mission is to support the collaborative development of Odoo features and promote its widespread use. The OCA is committed to providing resources and legal support to the Odoo community and ensuring that Odoo remains both open source and maintained for the public's benefit. You can find it at <https://odoo-community.org>.

Finding Odoo applications and modules

Odoo 12 is typically downloaded in a format that already comes with many applications and modules available for installation. However, there are many more modules in the online repository. If you are thinking about making any customization in Odoo, make sure that someone else hasn't done the work for you already by checking at: <https://apps.odoo.com/>.

Here, you can browse for modules by the application they are associated with and even filter out applications by the version number of Odoo you require. Starting with Odoo 9, there are paid applications in the library, as well as free offerings.

Getting the latest Odoo 12 release notes

The link to take you directly to the Odoo 12 release notes is: <https://www.odoo.com/odoo-12-release-notes>.

Downloading Odoo from GitHub

Odoo branches are maintained on GitHub. The primary Odoo project is located at <https://github.com/odoo/odoo>.

Locating resources on Ubuntu

While you can certainly work with Odoo under the Windows environment, most of the community agrees that it is better to run your Odoo server under Ubuntu. At the time of writing, the latest stable release version is Ubuntu 18.04. Here are some resources that can help you get started with Ubuntu.

Official Ubuntu website

If you are planning on working with Odoo and Ubuntu, you will want to begin by getting familiar with the official Ubuntu website at: <http://www.ubuntu.com/>.

Directly downloading the Ubuntu server or desktop

You'll find resources for downloading Ubuntu in various formats and configurations at: <http://www.ubuntu.com/download>.

Official Ubuntu documentation

The official Ubuntu website has a comprehensive set of documentation that is a great reference for troubleshooting your Odoo installation. You can find this documentation at <https://help.ubuntu.com/>.

Getting additional developer documentation

The Odoo framework provides a lot of power, but also comes with a fairly steep learning curve. While Chapter 16, *Comparative Analysis of Community versus Enterprise Editions*, of this book will help get you started, if you are serious about developing for Odoo, you will find the following resources valuable in your pursuit.

Getting quick access to Odoo installations using Odoo Runbot

Wouldn't it be great if you could try out a variety of Odoo builds and installations quickly and easily without going through all the time and complexity of an Odoo installation? Well, fortunately, the Odoo Runbot provides you with many different Odoo branches that you can connect to and actually review the build for yourself.

This resource divides the installations by version, build, and age of the version. Even better, you can get immediate access to the branches right on GitHub at: <http://runbot.odoo.com/>.

Finding Postgres resources

When you're supporting any ERP system, it is important to understand the underlying database architecture. Odoo 12.0 requires Postgres 9.3 or later as the backend database. Here is a list of resources that can help you maintain your Postgres server.

Official Postgres website

While Odoo handles many of the complex database management tasks you would have with other ERP systems, there will be times when the official Postgres website will be useful in maintaining your Odoo installation—find it at: <http://www.postgresql.org/>.

Note that Postgres is also commonly known as **PostgreSQL**; these terms are synonymous.

Downloading Postgres

The following links are for downloading Postgres in various configurations:

- <http://www.postgresql.org/download/>
- <http://www.postgresql.org/download/linux/ubuntu/>

Postgres documentation

The standard Postgres documentation will be useful when you need to perform operations on your Odoo database, at: <https://www.postgresql.org/docs/>.

Locating Python resources

Odoo modules are created in Python, a free and powerful programming language. Most module development will require you to at least become familiar with the basics of Python. Odoo requires Python version 2.7 (Python versions 3.x and above are not supported by the current builds of Odoo).

Official Python website

The link to the official website for Python is: <http://www.python.org/>.

Downloading Python

Depending on the platform you plan to install Odoo on, you may need to download Python before you can install Odoo, which you can do at: <http://www.python.org/download/>.

Python documentation

Writing modules or applications in Odoo requires basic knowledge of Python. The official Python documentation can be located at <http://docs.python.org/3/>.

Finding XML resources

Odoo views are designed and maintained in **Extensible Markup Language (XML)**. A basic understanding of XML will help you in customizing views, modifying search criteria, and managing workflows. Here is a list of XML resources that you can work with.

World Wide Web Consortium XML resource page

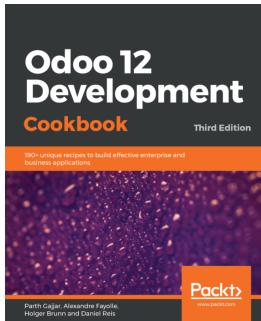
The official page for the XML specification is at: <http://www.w3.org/XML/>.

XML tutorials and documentation

If you are just beginning to learn about XML, make sure that you visit the excellent collection of tutorials and documentation at: <http://www.w3schools.com/xml/>.

Other Books You May Enjoy

If you enjoyed this book, you may be interested in these other books by Packt:

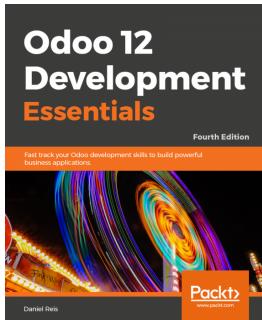


Odoo 12 Development Cookbook - Third Edition

Daniel Reis, Alexandre Fayolle, Et al

ISBN: 978-1-78961-892-1

- Develop a module in the Odoo framework and modify the existing flow of any application
- Build dynamic websites with Odoo CMS
- Create and modify backend JavaScript components in Odoo and POS
- Connect and access any object in Odoo via Remote Procedure Calls (RPC)
- Illustrate the different tools available in Odoo to implement business processes for your records
- Implement in-app purchase services
- Manage, deploy, and test an Odoo instance with the PaaS Odoo.sh
- Configure IoT Box to add and upgrade Point of Sale (POS) hardware



Odoo 12 Development Essentials - Fourth Edition

Daniel Reis

ISBN: 978-1-78953-247-0

- Manage Odoo server instances
- Create a new Odoo application from scratch using the most frequently used elements
- Develop new models and use inheritance to extend existing models
- Use ORM methods in the Odoo server and from external clients
- Create Kanban views using QWeb effectively
- Build custom web and website CMS pages
- Use external APIs to integrate Odoo with external applications
- Add automated tests and techniques to debug module business logic

Leave a review - let other readers know what you think

Please share your thoughts on this book with others by leaving a review on the site that you bought it from. If you purchased the book from Amazon, please leave us an honest review on this book's Amazon page. This is vital so that other potential readers can see and use your unbiased opinion to make purchasing decisions, we can understand what our customers think about our products, and our authors can see your feedback on the title that they have worked with Packt to create. It will only take a few minutes of your time, but is valuable to other potential customers, our authors, and Packt. Thank you!

Index

—
 __init__.py file 400
 __manifest__.py file 401, 402

A

Access Rights tab
 about 69
 email address field 69
 username, specifying 68
access rights
 in Odoo 199, 200, 201
account manager
 assigning 67
accounting
 configuration panel, using for 159, 160, 161, 162
 configuration, completing 165
accounts receivable process
 transactions, following through 169, 170, 172
accounts
 setting up 177, 178
 specifying, for product category 179, 180
actions 352, 370, 371, 372
activity
 scheduling 262, 263
advanced eCommerce product options
 about 340, 341
 accessory products 342
 alternative products 342
 checking out 345, 346, 347
 draft sales order, viewing in Odoo 344
 shopping cart, viewing 342
all-in-one Odoo installation, on Windows
 performing 23
API, Odoo
 used, for integrating applications 374

applications
 integrating, Odoo's API used 374
attendees
 adding, to meeting 96

B

Bill of Materials (BoM)
 about 445
 building 127, 128, 129
 routing orders, assigning to 142
 used, for creating kit 152, 153
breadcrumbs 62
business document 86, 87
business intelligence features 290, 291, 292, 293
business strategy
 CRM, using as 64
business
 chart of accounts, defining for 155
 search requirements, determining for 266

C

calls
 automating, with VoIP integration 437
 scheduling 94
chart of accounts
 about 155
 defining, for business 155
 information, obtaining on 163, 164
 reviewing 163
 setting up, for second company 225, 226, 227
Chatter feature 83
code
 writing, with Python 18
Community Edition, Odoo 9
company report configuration 378, 379, 380
complex units of measure
 defined units of measure, viewing 117

- handling 116
product, configuring to purchase in Dozen(s) 119
specific unit of measure, examining 118
- configuration panel**
used, for accounting 159, 160, 161, 162
- content management system (CMS)**
about 294, 323
DotNetNuke (DNN) 295
Drupal 295
Evoq 295
Joomla 295
WordPress 295
- Coordinated Universal Time (UTC)** 23
- CRM application**
installing 66
- custom dashboards**
creating, in Odoo 12 285, 286, 287, 288, 289
- custom field**
adding, to Odoo 357
- custom page layout**
creating 422, 423
- custom product**
examples 123
- custom search**
performing 279, 280, 281
- custom theme**
creating, in Odoo 419
custom page layout, creating 422, 423
Odoo theme, architecture 419
Odoo theme, creating with scaffolding 419, 420
- customer experience**
improving 65
managing 64
- Customer Relationship Management (CRM)**
about 8, 46, 417
using, as business strategy 64
- customer**
salesperson, assigning to 72
treating, like real person 64
- D**
- database manager**
accessing 267, 269, 270
- database**
backing up 355, 356
- country, specifying 35
creating 33
creating, with demonstration data 267
demonstration data, loading 35
email address, selecting 34
language, specifying 35
managing 36
name, selecting 34
password, selecting 34
phone number, specifying 35
restoring, in Odoo 356
- demonstration data**
in Odoo 12 270
- developer mode**
activating 353, 354
deactivating 355
- DotNetNuke (DNN) 295
- Drupal 295
- dynamic accounting reports** 433, 434, 435
- dynamic QWeb template report**
creating, t-if used 394
- E**
- eCommerce** 322
- eCommerce page**
product price, setting from 328
- eCommerce platforms**
about 323
Magento 323
Shopify 323
Volusion 323
Yahoo Small Business 324
- eCommerce site**
picture, adding to product 329, 330
product description 331
product, publishing 332, 333
- employee department**
about 232, 233
Coach field 235
Employee Private Information 236
Job Title field 234, 235
Manager field 235
- Employee Directory application**
installing 230
- employee**

adding 231
Enterprise Edition, Odoo 9
enterprise resource planning (ERP) 267
event
 scheduling 94, 95
Evoq 295

F

favorite filters
 saving 283, 284
fields
 about 352
 creating, in sale.order model 360, 361
filters
 using, in list views 275, 276, 277
Followup tab, Opportunity form
 address and contact information 79
 mailings 79
 marketing 80
form view
 editing 362, 363, 364, 365, 366

G

group inheritance
 in Odoo 197, 198
group rules
 in Odoo 202, 203, 204
grouping 277, 278, 279
groups
 in Odoo 195, 196, 197
 menus, defining for 198, 199

H

HR applications
 modular approach 229
HR Settings page
 about 237
 departments, managing 237

I

installation operating system
 selecting 19
internal report header
 modifying, QWeb used 386

international currencies 208, 209, 210, 211
internationalization
 in Odoo 204

J

Joomla 295
journal entries
 creating 182, 184

K

kanban view 273
key performance indicators (KPI) 439
keywords
 specifying, for website 319
kit
 creating, with BOM 153

L

language translation
 configuring 205, 206, 207, 208
leads
 about 90
 converting, into opportunity 93
 creating, in Odoo 91, 92
 in Odoo 12 90
leave management 241
leave request
 approving 245
 creating 242
 Department field 243
 Duration field 243
 Employee field 243
 Leave Type field 242
 Mode field 243
leave
 submitting, for approval 243
list views
 filters, using in 275, 276, 277
Long-Term Support (LTS) versions 21

M

Macintosh users 10
Magento 323
Maintenance application 445, 446

manufacturing applications
about 442
Maintenance 445, 446
maintenance requests, handling 445
PLM 443, 444, 445
quality 447, 448, 449

manufacturing orders
creating 122, 124, 125
creating, with routing 143, 144, 145, 146
creating, with work center 143, 144, 145, 146
order, delivering 122
product, manufacturing 125, 126
product, producing 122
workflow, defining for business 122

Manufacturing Resource Planning (MRP)
installing 123

meeting details
specifying 97

meetings
additional meeting options, specifying 98, 99
attendees, adding to 96
reminders, setting up for 98
scheduling 94
tags, specifying for 97

menus
adding, to website 310, 314
creating 373, 374
defining, for groups 198, 199
managing, on website 314

Microsoft Windows Odoo installation
selecting 19

mission statement
used, for driving design CRM system 65

mobile device
website, previewing on 309, 310

Model-View-Controller (MVC)
controllers, authoring 19
following 18
models, designing 18
views, rendering 18

models
about 352
viewing, in Odoo 358, 359

multi-company solution
implementing 228

multiple advanced searches
specifying 282

multiple companies
in Odoo 221

O

Object Relational Database Management System (ORDMS) 17

Odoo 12 Enterprise Accounting
examining 431

Odoo 12 Enterprise
launching, from Odoo runbot 430

Odoo 12
about 8
custom dashboards, creating 285, 286, 287, 288, 289
demonstration data 270
download link 22
Project Management 252

Odoo administration
basic considerations 186

Odoo API
connecting to 418
records, filtering 418
records, returning 418
search_read method, using 418

Odoo application and module repository
reference link 396

Odoo application
exploring 396

Odoo architecture
about 17, 352
actions 352
models 352
reference link 17
views 352

Odoo Community Edition
about 426
launching, from Odoo runbot 428, 429

Odoo configuration file
additional resources, finding on installing Odoo 30
admin password, modifying 30
database management tools, accessing 29
finding 397

installing 397
modifying 29
port numbers, modifying 29
Odoo configurations
 documentation, clearing of 187
Odoo CRM pipeline 67
Odoo database
 backing up 188, 189, 190
 restoring 190
Odoo eCommerce
 basics 326, 327
 installing 324, 325
Odoo Editions
 reference link 426
Odoo Enterprise Edition
 about 426
 options 436
 shipping connectors 441, 442
Odoo Enterprise interface
 examining 430, 431
Odoo free online trial
 accessing 11, 12, 13, 14
Odoo installation
 testing 26
Odoo interface 37
Odoo module
 building 396
 code, adding to *rush order filter* 413
 custom directory, specifying 398
 development environment, preparing 397
 directory, contents 400
 directory, exploring 396
 fields, adding to *sales order view* 405, 407
 files, creating 400
 files, editing 400
 filter, creating 414
 inheriting, from *Odoo Sales application* 404
 installation, troubleshooting 410, 411
 installing 408, 410
 Odoo model, extending in *silkworm.py* 403
 Python conventions 405
 used, for adding custom fields to model 403
 used, for adding filter to search view 412
Odoo Online instant access
 using 10
Odoo Online Pricing page
 reference link 16
Odoo Online
 using 33
 working with 9
Odoo releases 21
Odoo runbot
 reference link 427
 used, for comparing Odoo versions 427, 428
Odoo services
 checking 26, 27
 starting, in Ubuntu 28
 stopping, in Ubuntu 28
Odoo shop
 additional configuration options 333, 334
 order of products, modifying on store 335, 336
Odoo Studio
 advantages 450
 disadvantages 450
 Sales application, modifying 450, 451, 452
 using 450
Odoo theme
 architecture 419
 creating, with scaffolding 419, 420
 header, modifying 420, 422
Odoo Website Builder
 installing 298, 299, 300, 301, 303, 304, 305,
 306
 using, as CMS 296
Odoo, as CMS
 advantages 296
 limitations 297
Odoo, implementation strategy
 about 186
 development server 186
 documentation, clearing of Odoo configurations
 187
 focus, on business continuity 187, 188
 production server 186
 staging server 186
Odoo, installing on Ubuntu
 about 25
 sources.list file, modifying 25
Odoo.sh
 reference link 9

Odoo
about 8
as eCommerce platform 324
browser destination, checking 26
browser requisites 10
business intelligence features 290, 291, 292, 293
case study 32
Community Edition 9
custom field, adding 357
custom theme, creating 419
database, restoring 357
Enterprise Edition 9
implementing, with modular approach 32
installing, on Windows OS 21, 23
legal reports, examining 181, 182
management tips 26
mobile phone support 11
model, viewing 358, 359
OS option, selecting for 20
reports, organizing 387
search operations, customizing 366
searching 271, 273, 274, 275
sequences, managing 217, 219, 220
subscribing to 15, 16
tablet support 11
transactions, posting to accounts 165, 166, 168
trial version, using 15
troubleshooting 26
troubleshooting steps, reference link 28
upgrading 21
used, for creating web services 414, 415
used, for creating websites 414, 415
users, administering 191
using, without subscription fees 17
XML template, creating 416, 417

opportunity
about 90
contact information 78
converting, to quotation 90
creating 74
details, viewing of 82
Email and Phone fields 78
expected closing 78
in pipeline 76

Internal Notes area 78
lead, converting into 93
managing 85, 86
status, modifying 83
tags 78
Optical Character Recognition (OCR) 8
overall equipment effectiveness (OEE) 139

P

pages
adding, to website 310, 311, 312
Payment Terms field 408
PayPal
adding, as payment processor 348, 349
pipeline 73
pop-up panel, for opportunity
customer 75
expected revenue 75
opportunity 75
rating 76
Postgres
configuring, on Windows 23, 24
PostgreSQL
about 17
reference link 17
posting transactions
practicing 177
results, tracking 177
primary Odoo log file
finding 29
procurement 101
product category
accounts, specifying for 179, 180
practicing with 181
Product Lifecycle Management (PLM) application 443, 444, 445
product price
setting, from eCommerce page 328
product variants
setting up 337, 338, 339, 340
production
availability, checking 132, 133
confirming 131, 132
managing, by work orders 135, 136
products information

- Can be Purchased 49
- Can be Sold 48
- Cost Price 49
- Internal Reference field 49
- Product Name 48
- Product Type 49
- Sales Price 49
- products Invoicing page
 - entering 50
 - invoicing policy 51
- products
 - creating 48
 - entering 47
 - producing 133, 134
 - receiving 111
 - Sales tab, checking 50
 - saving 51
 - setting, as storable 130, 131
- project costs
 - calculating 264
- Project dashboard 254
- Project Management application
 - installing 253
 - real-world example 254
- Project Management
 - basics 253
- project tasks
 - additional tasks 260
 - completing 261
 - defining 258, 259
- project time
 - calculating 264
- project
 - creating 255, 256
 - stages 256, 257
- Purchase application
 - cost price, setting of product 105
 - delivery time, calculating 107
 - installing 102, 103
 - minimal quantity, setting 107
 - price, setting 108
 - product type 105
 - validity, setting 108
 - vendor, setting up 103, 104
 - vendors, assigning to product 105
- purchase order
 - confirming 110
- purchase quotation
 - creating 108, 109
 - products, adding to request for quotation 109
 - RFQs, printing 110
 - status, updating 110
- purchasing process
 - about 100
 - in different currency 212
 - invoice, settling 102
 - product, receiving 102
 - purchase orders, generating 101
 - quotations, generating 101
 - vendor, setting up 101
 - warehouse locations, setting up 101
- Python
 - code, writing with 18
 - download link 374
 - reference link 18
- Q**
 - Quality application 447, 448, 449
- quotation sales order
 - line items, entering on 55
 - saving 57
- QWeb report
 - creating, in Odoo 393
- QWeb template engine 377
- QWeb templates 393
- QWeb
 - used, for modifying internal report header 386
- R**
 - Recruitment Process application
 - about 245
 - employees, hiring 250
 - employment application, creating 248
 - installing 245, 246
 - recruiting, for new job position 246, 247
 - reminders
 - setting up, for meeting 98
 - reports
 - organizing, Odoo used 387
 - paper format, specifying for 381, 382

- simple change, making 383, 384, 385
skills, required for modification 377
types 388
- Request for Quotation (RFQ) 108, 110
- reverse Bill of Materials (BoM)
using 151
- routing orders
assigning, to bill of materials (BoM) 142
creating 140, 141
- routing, creating based on product variants
about 147, 149
creation, controlling of variants 149
size attribute, adding 149, 150, 151
- routing
manufacturing order, creating with 143, 144, 145, 146
- ## S
- sale.order model
field, creating in 360, 361
- Sales application settings
about 69
language 71
Manager setting 70
messaging and social, notification management 71
Signature section 71
timezone 71
User all documents 70
User own documents only 69
user password, setting manually 71
- Sales application
company data, configuring 38, 39
customer accounting, editing 47
customer Sales & Purchase page, editing 45, 46
customer, specifying 43
data, entering into Odoo form 44, 45
first customer, creating 42, 43
installing 36, 37
modifying, with Odoo Studio 450, 451, 452
multiple addresses, using 45
multiple contacts, using 45
payment options, specifying 40
quotation layout, selecting 40
quotation, sending 41, 42
- Sales order form 388, 389, 390, 391, 392
sales order workflow
about 58
Cancel 59
Confirm Sale 59
Print 58
sales order workflow 57
Send by Email 58
- sales order, field
Description 56
ordered quantity 56
Product field 55
taxes 56
unit price 56
- sales order
customer, selecting 53
entering 52, 53
payment terms 54
Validity date 54
- sales representative
assigning 67
- sales stages
about 80, 81
Activities menu 82
modifying 87, 88
- sales-related enhancements
about 437
calls, automating with VoIP integration 437
Subscriptions application 438, 440
- sales
closing 89
invoice options 60
invoicing 60, 61, 62
transactions, following through 169, 170, 172
- salesperson
assigning, to customer 72
- search engine optimization (SEO) 317
- search operations
additional fields, specifying 366
customizing, in Odoo 366
search view, editing 368, 369, 370
- search requirements
determining, for business 266
- search view
Odoo module, used for adding filter 412

searching, Odoo
about 271, 273, 274, 275
custom search, performing 279, 280
favorite filters, saving 283, 284
filters, using in list views 275
information, grouping 277, 278
multiple advanced searches, specifying 282

second company
Chart of Accounts, setting up for 225, 226, 227
setting up 221, 223, 224

sequences
managing, in Odoo 217, 219, 220

shipping connectors
in Odoo Enterprise 441, 442

Shopify 323

Silkworm 32, 116

stock valuation
analyzing 135

Subscriptions application 438, 439, 440

superuser account 34

supplier price lists
managing 213, 214, 215, 217

T

t-if
used, for creating dynamic QWeb template report 394

tags
specifying, for meeting 97

task completion
monitoring 263, 264

themes
modifying 315, 317

Timesheets application 238, 239, 240

transactions
viewing, created by invoice validation 173, 174, 175, 177

tuple 414

U

Ubuntu Odoo installation
selecting 20

Ubuntu
about 19
features 20

Odoo services, checking 28
Odoo services, starting 28
Odoo services, stopping 28

unbuild orders
using 151

user preferences
managing 193, 194

users
administering, in Odoo 191
configuring, to view Community Edition
accounting options 155, 158
selecting, to administer 191, 192

V

vendor bill
creating 111
paying, without purchase order 112, 113, 114, 115

vendor
designating, as Individual or Company 104
establishing 106
product code, designating 107
product name, designating 107
setting up 101

views 352

VoIP integration
used, for automating calls 437

Volusion 323

W

web page
saving 307, 308

web services
creating, Odoo used 414, 415

website
backing up 297
creating, Odoo used 414, 415
description, creating for 320
frequently asked questions web page, creating 313

keywords, specifying for 319

menus, adding 310, 315

menus, managing on 314

pages, adding to 310, 311, 312

previewing, on mobile device 309, 310

promoting 317
title, creating for 320

Windows OS
Odoo, installing on 21, 23

Windows
Postgres, installing on 23, 24

Wkhtmltopdf (WebKit HTML to PDF) installation
issues 381
URL 381

WordPress 295

work center, general information
about 138
Capacity 139
overall equipment effectiveness target 139

time after production 139
time before production 139
Time Efficiency 139

work center
creating 137, 138
manufacturing order, creating with 143, 144, 145, 146

X

XML template
creating 416, 417

Y

Yahoo Small Business 324