

深入系统介绍Odoo开发，支持Odoo11和Odoo12版本

实践性强，书中结合bug管理系统和培训机构管理系统两个案例来讲解理论知识
随书附带源码下载链接，让您“一书在手，Odoo开发全懂”

刘金亮 / 著

Odoo

快速入门与实战

Python开发ERP指南

Odoo
development introduction and practice
Python developing ERP Guidelines



机械工业出版社
China Machine Press



Odoo development introduction and practice
Python developing ERP Guidelines

Odoo

快速入门与实战

Python开发ERP指南

刘金亮 / 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Odoo 快速入门与实战: Python 开发 ERP 指南 / 刘金亮著. —北京: 机械工业出版社, 2019.4

ISBN 978-7-111-62530-8

I. O… II. 刘… III. 企业管理 - 应用软件 - 软件开发 IV. F272.7

中国版本图书馆 CIP 数据核字 (2019) 第 068122 号

Odoo 快速入门与实战: Python 开发 ERP 指南

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 张志铭

责任校对: 殷虹

印刷: 北京瑞德印刷有限公司

版次: 2019 年 5 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 18.75

书号: ISBN 978-7-111-62530-8

定价: 79.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

Odoo 是什么

Odoo 是一个开放源码且功能完善的企业级管理软件平台，它不仅提供了 ERP 常规模块，而且还允许使用人员进行灵活的实施配置。对于前台配置解决不了的需求，还可以通过 Odoo 提供的平台进行二次开发。

我曾在 SAP 领域工作了 8 年，参与和主导过的项目金额超过千万元，因此我能深刻体会到一个好的 ERP 产品在实施过程中是多么重要。目前国内虽然也有不错的 ERP 产品，但全是靠厂商及其合作伙伴来实施并进行二次开发，如果甲方想要自己培养技术人员则非常困难，很少有技术人员愿意将自己的职业发展方向捆绑在一个封闭又落后的技术框架上。这时候，Odoo 开源及其使用 Python 开发的优点就显现了出来，不仅企业可以免费配置使用，而且还可以方便地找到二次开发人员，甚至也可以找到具有学习热情的内部人员。毕竟 Python 的魅力及其在互联网领域越来越广的应用，决定了以后 Odoo 开发的难度会越来越低。

Odoo 良好的可扩展性可以媲美任何主流的 ERP 产品，其允许按照模块实现新的需求；在二次开发的过程中，如果要修改已有功能的话，也不需要直接修改源码，只需使用继承的方式进行增强和扩展，或者在子类中修改字段或方法。

Odoo 从其前身 OpenERP 开始，就受到了来自社区的活跃更新的支持；目前，Odoo 有数千个社区模块，几乎所有的主题都能找到相应的社区模块，这也反过来进一步吸引了更多的人员参与到 Odoo 的社区中来。

谁适合阅读本书

本书作为 Odoo 的开发指南，按照由简入难的步骤对 Odoo 的开发进行了全面介绍，不仅

可以让你快速入门 Odoo，同时也可以作为你进行项目实战的不错的参考资料。

本书适合于想使用 Odoo 进行开发的开发者，在阅读本书前甚至不需要具备 Python 基础编程知识，因为书中专门有一章（第 4 章）对 Python 进行了介绍。书中所列案例使用的代码简单明了，对于使用中会遇到的一些“坑”也给出了特别说明。

同时，本书也适合于 Odoo 的关键用户和最终用户，如果你在拥有 Odoo 系统的使用经验的前提下，还希望提升自己的开发技能，以便日后在工作中能够完成更为复杂的需求实现，那么本书将是非常好的参考用书。

如果你已经是 Odoo 开发人员，那么可以将本书作为不同版本之间的对比用书，毕竟从 Odoo8 到目前的 Odoo12 有了较大的变化。

最后，在校大学生也可以使用本书制作自己的案例、编写论文，甚至直接作为未来工作的编程平台。Odoo 在国内的市场越来越大，特别是非一线城市对于 Odoo 人才的需求也越来越大，如果作为学生的你不想将自己未来的职场定位在北上广深，那么 Odoo 将是一个非常不错的选择。

如何阅读本书

本书在逻辑上分为三篇：基础篇、技术研发篇和实战篇。基础篇（第 1～5 章）包括了 Odoo 的介绍及安装配置、Python 基础和 PostgreSQL 数据库介绍。不了解 Odoo 的读者请认真阅读第一篇，以免在后面的章节中出现不了解的名词。特别是没有 Python 基础的读者，最好能够认真学习第 4 章。

技术研发篇（第 6～14 章）则专注在 Odoo 的技术范畴内，包括了从视图到 ORM 和安全配置的全方位介绍，并且都是以 bug 管理系统的实现为示例来进行讲解的，读者在学习本篇内容时应边参考本书边动手实现。如果读者已经有 Odoo 基础，则可以直接下载本书的源码，在遇到不理解的内容时再查找相应知识点。

而对于实战篇（第 15～19 章），其内容从培训机构的管理系统的需求分析开始，再到划分模块，最后到具体的逻辑和视图的实现，非常适合有充足时间的初学者，特别是学生。

读者在阅读本书时，要注意源码的使用，具体的源码下载方式会在后续的“源码与勘误支持”部分提供。

在阅读本书的过程中，读者有任何疑问都可以访问作者网站 www.scott-odoo.com，以进行提问和交流。

源码与勘误支持

书中提供了核心源码，而且为了便于阅读特意增加了行号。需要读者注意的是，因为排版的问题，可能将原本属于一行的代码自动换行成两行，所以请读者以行号为准，凡是不带行号的代码说明都属于上一行。

本书的示例代码以及实战项目的源码可以到 GitHub 下载，下载地址为：

<https://github.com/ScottAI/Odoo-Python-ERP->

实战项目同时支持 Odoo11 和 Odoo12，Odoo 端的下载地址如下：

<https://github.com/ScottAI/-Odoo--->

实战项目的小程序端下载地址为：

<https://github.com/ScottAI/lexuebao-odoo-wechat>

由于作者的水平有限，加之撰写时间仓促，难免有错误或不合理之处，欢迎读者指正，你可以通过以下任何一种方式联系到作者：

作者的 QQ：784680552

作者的邮箱：wkdjl@163.com

作者的个人博客：www.scott-odoo.com

Contents 目 录

前言

第一篇 基础篇

第1章 Odoo简介 2

- 1.1 Odoo 发展历程 2
- 1.2 Odoo 与主流 ERP 系统的对比 3
- 1.3 总结 4

第2章 安装与配置 5

- 2.1 在 Ubuntu 上安装 Odoo 5
 - 2.1.1 创建 Odoo 用户并授权 5
 - 2.1.2 更新依赖包并使用 git 克隆 Odoo 源码 6
 - 2.1.3 安装 Odoo 依赖包及 PostgreSQL 6
 - 2.1.4 启动 Odoo 服务 7
- 2.2 在 Windows 上安装 Odoo 8
 - 2.2.1 准备工作 9
 - 2.2.2 环境配置 9

- 2.3 使用和创建 Odoo 数据库 10
 - 2.3.1 创建 Odoo 数据库 10
 - 2.3.2 语言选择 11
- 2.4 服务器配置 13
 - 2.4.1 Odoo 服务器配置文件 13
 - 2.4.2 远程开发 15
- 2.5 安装模块 16
 - 2.5.1 配置模块插件路径 16
 - 2.5.2 更新应用列表 16
- 2.6 总结 17

第3章 Odoo应用概览 18

- 3.1 整体架构 18
- 3.2 什么是开发者模式 19
- 3.3 开发者模式应用 20
 - 3.3.1 模块需求简介 21
 - 3.3.2 向已有模型新增字段 21
 - 3.3.3 新建菜单 25
 - 3.3.4 新建自定义模型 28
 - 3.3.5 新建视图 29
 - 3.3.6 安全性配置 30
- 3.4 base 模块简介 32

6.4.2	创建列表视图	71	7.6	总结	97
6.4.3	业务文档表单视图	72	第8章	文件相关数据操作	98
6.4.4	搜索视图	74	8.1	外部 ID	98
6.4.5	视图继承	74	8.2	导入导出数据	101
6.5	业务逻辑	76	8.2.1	导出数据	101
6.6	安全性配置	77	8.2.2	导入数据	102
6.6.1	访问控制	77	8.2.3	CSV 中关联字段的导入	103
6.6.2	网页和控制器	79	8.3	模块数据	104
6.7	总结	80	8.3.1	演示数据	104
第7章	自建应用进阶	81	8.3.2	XML 数据文件	105
7.1	模型	81	8.4	总结	108
7.1.1	模型属性	83	第9章	ORM API基础	109
7.1.2	模型与 python 类	83	9.1	常用装饰器	109
7.1.3	瞬态和抽象模型	84	9.1.1	记录集处理装饰器	109
7.1.4	透视已有模型	84	9.1.2	专有目的装饰器	110
7.1.5	模型扩展	85	9.2	ORM 内置方法	112
7.2	字段	86	9.2.1	模型写入数据的方法	112
7.2.1	创建字段	86	9.2.2	使用 RPC 的网页端方法	113
7.2.2	字段常用属性	87	9.3	数据导入导出方法	114
7.2.3	特殊字段	88	9.4	通信 API	114
7.3	模型关系与复杂字段	88	9.4.1	邮箱和社交 API	115
7.3.1	关系字段	89	9.4.2	消息子类型	116
7.3.2	模型关系	89	9.4.3	发布消息	116
7.3.3	one2many 的逆向关系	91	9.4.4	增加关注者	117
7.3.4	层级结构关系	91	9.5	总结	117
7.3.5	使用引用字段的动态关系	92	第10章	ORM: 业务逻辑处理	118
7.4	更多模型继承机制	93	10.1	向导	118
7.4.1	使用原型继承	93	10.1.1	创建向导	118
7.4.2	使用委托继承	94			
7.5	视图	94			

10.1.2	向导模型	120	第12章 后端视图	151	
10.1.3	向导 form	120	12.1	菜单项和窗口动作	152
10.1.4	向导业务逻辑	123	12.2	表单视图	154
10.1.5	异常处理	124	12.2.1	表头	156
10.2	更多 ORM API 用法	126	12.2.2	工作表	157
10.2.1	shell 命令	126	12.3	视图字段	160
10.2.2	服务器环境	127	12.4	按钮	162
10.2.3	事务和底层 SQL	128	12.5	看板视图	164
10.3	对记录集的操作	130	12.5.1	看板视图基本用法	166
10.3.1	查询模型	130	12.5.2	看板卡片布局	169
10.3.2	单例模式	131	12.5.3	看板卡选项菜单	171
10.3.3	写记录集	131	12.5.4	看板视图动作	172
10.3.4	日期和时间	132	12.6	其他类型的视图	172
10.3.5	操作记录集	133	12.6.1	列表视图	172
10.3.6	关系字段	135	12.6.2	搜索视图	174
10.4	总结	135	12.6.3	日历视图	175
			12.6.4	图形和数据透视图	177
第11章 创建网站		136	12.7	总结	179
11.1	第一个页面	136	第13章 QWeb	180	
11.1.1	创建新模块插件	136	13.1	客户端 QWeb	180
11.1.2	Hello World	137	13.1.1	JavaScript 表达式	181
11.1.3	Web 功能扩展	138	13.1.2	动态替换属性	181
11.1.4	Hello CMS	141	13.1.3	循环指令	183
11.2	前端页面渲染	142	13.1.4	条件指令	184
11.2.1	CSS 和 JavaScript	142	13.1.5	渲染值	184
11.2.2	列表页面	143	13.1.6	设置变量值	185
11.2.3	表单页面	145	13.1.7	复用模板	185
11.3	权限控制	148	13.1.8	CSS 和 JavaScript	186
11.4	总结	150	13.2	报表	187

13.2.1	安装 wkhtmltopdf	187	15.3.1	模块详细设计	212
13.2.2	创建商业报表	188	15.3.2	git 与环境	213
13.3	服务器端 QWeb	188	15.4	总结	216
13.3.1	QWeb 报表模板	188	第16章 系统核心模块实现		217
13.3.2	报表展示数据	190	16.1	科目及教师	217
13.3.3	基于客制化 SQL 的 报表	190	16.1.1	模型	218
13.4	总结	192	16.1.2	视图	220
第14章 与外部系统的集成		193	16.2	班级和学生	224
14.1	Python 客户端调用	193	16.2.1	模型	225
14.2	客户端应用程序开发	196	16.2.2	视图	228
14.3	ERPpeek 客户端	199	16.3	课程和计划	229
14.4	OdooRPC 库	200	16.3.1	模型	230
14.5	总结	201	16.3.2	视图	235
	第三篇 实战篇		16.4	菜单	236
第15章 Odoo培训行业管理 系统设计		204	16.5	安全	238
15.1	开发背景	204	16.6	总结	239
15.1.1	培训机构一般情况介绍	205	第17章 费用及CRM模块的实现		240
15.1.2	培训机构管理中出现的 主要问题	206	17.1	费用模块	240
15.1.3	系统开发的必要性 和可行性	207	17.2	CRM 模块	244
15.2	系统功能设计	207	17.3	总结	246
15.2.1	系统业务流程分析	207	第18章 系统小程序端实现		247
15.2.2	系统功能需求分析	209	18.1	小程序开发简介	247
15.3	模块及环境	211	18.1.1	小程序的 Hello World	248
			18.1.2	小程序代码结构	249
			18.2	Odoo 与小程序的集成	252
			18.2.1	小程序登录机制	252
			18.2.2	小程序鉴权	255
			18.2.3	小程序登录 Odoo	257

18.3	模型	260	19.3	系统部署	275
18.3.1	模块整体架构	260	19.3.1	可用的预建包	275
18.3.2	重要模型	261	19.3.2	安装依赖	276
18.4	总结	265	19.3.3	准备专用系统用户	277
第19章	测试与部署	266	19.3.4	从源代码安装 Odoo	277
19.1	系统测试	266	19.3.5	设置配置文件	278
19.1.1	单元测试	266	19.3.6	多进程运行	279
19.1.2	配置测试	267	19.3.7	将 Odoo 设置为系统 服务	279
19.1.3	编写测试用例	268	19.3.8	创建系统服务	280
19.1.4	测试异常	269	19.3.9	创建 Upstart/sysvinit 服务	281
19.1.5	运行测试	270	19.3.10	命令行检查 Odoo 服务	282
19.1.6	YAML 测试	270	19.3.11	使用反向代理	282
19.2	系统调试	271	19.3.12	将 Nginx 设置为反向 代理	283
19.2.1	开发工具	271	19.3.13	设置 HTTPS	284
19.2.2	服务器端开发选项	271	19.3.14	Nginx 优化	285
19.2.3	调试	272	19.3.15	长轮询	286
19.2.4	Python 调试	272	19.3.16	服务器和模块更新	287
19.2.5	会话调试示例	273	19.4	总结	288
19.2.6	交互性 Python 调试器	274			
19.2.7	打印信息和记录日志	274			
19.2.8	“杀死”运行中的进程	275			

基础篇

Odoo 是当今最主流的免费管理系统和 ERP 系统框架，要想在企业内部更好地使用它，进行二次开发是非常有必要的，可是并非所有的情况都需要通过二次开发来完成，同时我们也应该知道，任何东西的应用都是有学习成本的，特别是企业级的应用系统。所以，本篇的主要目的是向读者介绍什么是 Odoo，如何安装 Odoo，如何通过配置来完成简单的需求，以及 Python 和 PostgreSQL 数据库的基础。

如果读者已有 Python 编程基础则可直接

跳过第 4 章，对于有其他语言编程经验的读者，仅通过学习第 4 章就可以掌握 Odoo 所用到的 Python 基础知识点。PostgreSQL 数据库则可以在后面的实战部分完成后再进行学习，Odoo 提供了良好的 ORM，而我们一般不需要直接访问 PostgreSQL 数据库。

在学习本篇的过程中，建议读者一定要多动手，如果仅仅是想了解 Odoo，则可以直接阅读第 3 章；要想成为真正可以进行开发的工程师，则需要边学习边练习。

Odoo 简介

Odoo 是一个完整的中小企业 ERP 系统，其中的模块包括管理会计、财务会计、库存管理、销售、采购管理、人力资源管理、项目管理等，总共 700 多个。Odoo 遵循通用公共许可证 (GUN)，并且源码开放，个人和企业都可以在遵循 GUN 协议的情况下使用 Odoo。

Odoo 因为其高度模块化和开源免费的特点，目前在世界范围内得到了高速的发展，是 SAP、Oracle 以外应用最广泛的管理系统，后续我们将从发展历程和对比分析的角度来介绍一下 Odoo。

1.1 Odoo 发展历程

结合国外科学数据维基网站 (www.limswiki.org) 提供的数据，我们可以梳理出 Odoo 的发展历程。Odoo 之前的名称是 OpenERP，它由当时还是学生的 Fabien Pinckaers 在 2002 年 5 月开发的一个小项目 TinyERP 发展而来。当 Pinckaers 在 2005 年完成学业时，该项目已经发展到足够大的程度，Pinckaers 引入了一个合作伙伴并传播了 TinyERP 的开源性质。然而，到 2008 年 7 月，计划开始改变项目的名称和重点。于是，Launchck 项目于 2008 年 7 月 24 日由 Pinckaers 启动，到 2008 年 10 月，该项目向 OpenERP 的过渡已经完成。

2010 年 2 月，OpenERP 从 Sofinnova Partners 手中获得了 300 万欧元的风险投资，用于加速公司的战略发展，包括加强管理团队、增加新 SaaS OpenERP 产品 (www.odoo.com) 的分销以及增强它的国际化。

2014 年 5 月 15 日，Fabien Pinckaers 宣布 OpenERP 更名为 Odoo。Pinckaers 引用了越来越多的应用模块和产品的“开箱即用集成度”，这些产品已从企业资源规划平台得到大幅度提升。Pinckaers 说：“我们需要一个没有约束的品牌。我们想要一个品牌，允许我们发布销售、会计、库存、采购等 ERP 应用程序。但该品牌也应该适合我们的最新应用，如网站

建设者、电子商务、销售点或商务智能。”

开发团队声称“Odoo”这个名字没有任何意义。有种解释是Odoo的域名已经被Pinckaers团队控制，而且团队发现“公司估价与其名称中‘O’的数量存在着直接的关联。”所以最终选择使用Odoo作为新产品名。实际上www.odoo.com这个域名从2009年到2010年一直用于分享OpenERP当时的新型软件即服务（SaaS）产品的服务。这也说明Odoo产品的模块范围远大于传统ERP领域。

发展到目前，除了免费开源的社区版以外，Odoo还提供了收费的企业版。目前该公司有员工约500人，服务用户约370万人。

1.2 Odoo与主流ERP系统的对比

首先，从国际市场占有率来看，最知名的ERP系统肯定是目前市场占有率第一的SAP，不过可以这样说，Odoo与SAP并不是竞争关系，SAP的客户主要来自于世界500强等大型国际化公司，而Odoo则主要应用于中小型企业。

从产品的功能特性来说，Odoo的特点是灵活、便于二次开发以及免费开源。而SAP则是具有强大的可配置性和丰富的行业解决方案，二次开发的学习成本非常高，需要使用其特有的ABAP语言。

从国内的ERP系统市场占有率来分析，Odoo的市场份额明显更大，因为与国内的ERP产品相比，Odoo的优势比较明显。

国内ERP系统基本上都是以某个行业或模块为核心来进行客户定位、功能设计与市场营销的，比如金蝶、用友主打财务，鼎捷软件（神州数码）主打生产，其标准产品都是可以直接使用的。Odoo本质上是一个用于快速开发的基于Python的Web架构ERP系统框架，而并非行业软件。

Odoo中自带的各个模块并不是提供给大家直接使用的，绝大多数情况下它们都无法直接使用。原因很简单，其主流程虽然是完善的，但并不具备分支流程和异常流程，要想正常使用大都需要进行二次开发。而Odoo的精妙之处正在于此，大多数公司的主流程都大同小异，直接使用原生模块即可，分支流程和异常流程需要基于原生模块对其进行定制。Odoo框架的设计初衷就是让开发人员用最少的代码实现最多的功能，甚至不用代码仅用配置即可实现。因此其用于开发管理系统的开发速度可以说是其他框架或语言的10倍都不止。

综上所述，Odoo本质上其实就是一个快速实现各种定制化管理系统的框架工具，而且其本身也形成了一定的社区和生态。

那么在具体的产品上，Odoo具有哪些优势呢？下面就来讲解如下。

- ❑ **电子商务。**虽然SAP提供了电子商务功能，但与Odoo提供的企业网站、电子商务、企业论坛以及电子商务与ERP系统的完美结合这几点对比来看，SAP还是有其先天弱勢的，SAP需要大量的实施工作。而国内的ERP产品基本上都依赖于定制开发，

费时费力而且 bug 比较多。

- **消息及邮件系统。**这一点是 Odoo 最鲜明的特色，其在很大程度上实现了 OA 的大部分功能。该项功能可以完成关注信息的自动推送，内部信息的发送和群发，消息系统和企业邮箱的继承，以及与日历的集成等操作。
- **客户关系管理。**虽然很多 ERP 系统都提供 CRM，但 Odoo 的 CRM 却有其自己的特色。客户的访谈和日历是集成的、会议也是集成的，以便记录客户的各种信息；所有的单据都提供了附件上传功能，以便上传自己记录的文件。
- **多语言、多公司、多货币。**特别是与国内的 ERP 产品相比，Odoo 在这一点上的优势非常明显，Odoo 可以根据个人用户的设置将界面自动翻译成用户的本地语言，也可以将多种货币兑换成公司的本币。这两项功能在国内的 ERP 产品中都是没有的，多公司则是允许一个 ERP 系统内有多个公司。

综上所述，可以看到 Odoo 在中小企业范围内是非常优秀的 ERP 产品，如果企业恰好没有 OA 系统，或者恰好从事外贸等需要多语言的业务场景，那么 Odoo 无疑是企业的首选。

1.3 总结

本章介绍了 Odoo 是什么以及 Odoo 是如何发展而来的，并将其与国内外主流 ERP 系统进行了对比，分析了 Odoo 的主要优势。

在了解了 Odoo 的特点以后，你是不是对进一步了解和学习 Odoo 充满期待呢？那么接下来我们就了解 Odoo 的安装和配置。

安装与配置

本书的编写宗旨是以实用为主，也希望读者在使用本书时能够在实际环境中边操作边参考本书。本书编写案例所使用的 Odoo 环境是安装在云服务器上的，使用的操作系统是 Ubuntu16.04，建议读者也使用该操作系统。如果身边没有合适的操作系统，又觉得在 Windows 系统上安装虚拟机太消耗性能，那么建议租一个云服务器作为学习使用，其性价比非常高，也符合目前的主流工程应用方向。

当然，如果硬件配置较高的话，那么在 Windows 上安装 Linux/Unix 虚拟机，然后在虚拟机上进行应用也是可以的。

这里我们还是建议大家首选 Debian/Ubuntu 系统，虽然 Odoo 在多种操作系统上都可以运行，但 Odoo 的开发团队一直将 Debian 作为参考部署平台，而且其 SaaS 应用也是基于 Debian 的。这就意味着，如果选择 Debian 或 Ubuntu，那么从网上查找相关资料将会更容易。

2.1 在 Ubuntu 上安装 Odoo

2.1.1 创建 Odoo 用户并授权

在 Ubuntu 上安装 Odoo 时，首先需要注意一点的是：不能在 root 用户下安装 Odoo。所以第一步，我们需要在操作系统上新创建一个用户，新建用户的命令如下：

```
1. $ sudo adduser odoo
```

然后，编辑文件 /etc/sudoers 为 Odoo 授权，修改文件命令如下：