
本书下载地址：

<http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1>

本书是《OpenERP 应用和开发第六章》，上述坛子是本书的连载基地，
之前和之后的章节请到上述坛子里找。

OpenERP 应用和开发基础 (第六章)

作者：老肖

msn: saiwai76@hotmail.com

mail: xiangfu.xiao@gmail.com

QQ: 1417063315

Shine-it 注册名：NewZN

OpenERP中国本地化项目： <http://code.google.com/p/hornerp/>

更新日期：

2009 年 12 月 24 日，完成第一章

2009 年 12 月 28 日，添加第二章

2009 年 12 月 30 日，添加第三章

2010 年 01 月 05 日，添加第四章

2010 年 01 月 11 日，添加第五章

2010 年 01 月 20 日，修改第三章

2010 年 01 月 21 日，添加第六章

目录

第一章	简单进销存案例	1
第二章	OpenERP开发入门	2
第三章	工作流和报表	3
第四章	OpenERP的权限设置	4
第五章	OpenERP翻译方法	4
第六章	OpenERP模块开发	5
第一节	OpenERP模块结构	5
一、	安装示例代码	5
二、	请假模块结构	6
第二节	理解OpenERP的代码	7
一、	对象定义	7
二、	视图、动作及菜单定义	9
三、	工作流和报表定义	10
四、	权限定义	10
第三节	服务端动作(Server Action).....	10
一、	配置Server Action	10
二、	配置OpenERP Mail	11
三、	测试通知Mail	12
四、	更多Server Action	13
第七章	超强的继承功能	15
第八章	基于XML-RPC的集成.....	15

第一章 简单进销存案例

下面是本章的目录，本章内容请到坛子里找，坛子地址：

<http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1>

第一节	进销存业务流程	1
一、	采购流程	1
二、	销售流程	1
三、	系统功能	2
第二节	openERP 系统安装	2
一、	Windows 版安装	3
二、	初次安装者常见问题	6
三、	中文化包的导入	7
第三节	从源码安装 openERP	8
一、	Linux 版安装	8
二、	Windows 版安装	13
第四节	模块配置	14
一、	财务设置	14
二、	设置业务伙伴及产品信息	17
三、	确认库位和仓库	20
第五节	测试采购、销售流程	22
一、	采购流程	22
二、	销售流程	28

第二章 OpenERP 开发入门

下面是本章的目录，本章内容请到坛子里找，坛子地址：

<http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1>

第一节	基本概念	32
第二节	实例演练	34
一、	创建请假单对象	34
二、	创建视图	35
三、	创建菜单和系统动作（Action）	37
四、	测试	39
五、	视图编辑器	40
第三节	深入数据库	41

第三章 workflows和报表

下面是本章的目录，本章内容请到坛子里找，坛子地址：

<http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1>

第一节	理解工作流（Workflow）	3
第二节	实作工作流	4
一、	修改对象和视图	4
二、	创建工作流、活动和迁移	6
三、	测试工作流	9
第三节	OpenERP 报表开发的预备知识	9
一、	报表开发工具	10
二、	XSL 简介	11
第四节	OpenERP 报表简介	11
一、	基于 sxw 或 odt 模板的报表	11
二、	基于 xsl+xml 模板的报表	13
第五节	报表中文化解决方法	13
一、	让报表显示汉字	14
二、	让报表说中文	15
第六节	实作报表	15
一、	编写 RML 文件	15
二、	编写报表对象代码及 XML 文件	16
三、	测试	17

第四章 OpenERP 的权限设置

下面是本章的目录，本章内容请到坛子里找，坛子地址：

<http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1>

- 第一节 OpenERP 权限机制简介 4
- 第二节 组的权限设置方法 5
- 第三节 Domain 条件表达式 8
 - 一、 Domain 条件表达式的应用场合 8
 - 二、 Domain 条件表达式写法 9

第五章 OpenERP 翻译方法

下面是本章的目录，本章内容请到坛子里找，坛子地址：

<http://www.shine-it.net/viewthread.php?tid=756&extra=page%3D1>

- 第一节 资源术语翻译 5
- 第二节 通用术语翻译 7
- 第三节 批量翻译或纠正翻译 8
- 第四节 OpenERP 国际化原理浅析 11

第六章 OpenERP 模块开发

一个真实的请假流程，通常要考虑如下要求：

- R1：普通员工只可以看见自己的请假单。
- R2：只有主管才可以批准请假单。
- R3：主管不应该看见草稿状态的请假单。
- R4：主管只可以看见自己部门的请假单。
- R5：主管可以查询自己部门的非草稿状态的所有请假单。
- R6：请假申请时发 Mail 通知主管，有待审批请假单。

本章继续前面章节的例子，开发一个完整的请假模块，实现上述要求。

第一节 OpenERP 模块结构

本节先安装示例代码，演示代码功能，而后介绍示例代码结构，解释 OpenERP 模块的组成部分及各部分的作用。

一、安装示例代码

将本章附件代码中的”qingjia”目录拷贝到 OpenERP 的安装目录下的 addons 下，点击菜单：管理员 → 模块管理 → Update Modules List 。检查模块更新后，在未安装模块下会出现新模块”qingjia”，安装它。请假模块在系统中增加菜单：

请假申请
 请假单审批
 所有请假单
 我的请假单

其中：

- 我的请假单：显示当前用户的所有请假单，包括未提交、被拒的、已批的、待批的所有请假单。所有员工都可以看见该菜单。
- 请假单审批：显示本部门的待批请假单，只有部门经理才能看见该菜单，普通员工看不见该菜单。
- 所有请假单：显示本部门的所有已提交的请假单，包括待批的、被拒的、已批的。只有部门经理可以看见该菜单。

请假模块还增加权限组 “qingjia / department manager”， workflow 角色 “部门经理”，以及下述用户：

登录名	登录密码	用户名	主管	权限组	角色
yg01	123	员工 1	经理 1		
yg02	123	员工 2	经理 2		
jl01	123	经理 1		department manager	部门经理
jl02	123	经理 2		department manager	部门经理

以员工身份登录系统，只能看见“我的请假单”菜单。以经理身份登录系统，三个菜单都能看见，但只能看见自己部门的请假单。员工在“我的请假单”中创建请假单，送批。而后，虽然画面上有“批准”

和“拒绝”按钮，但员工不拥有“部门经理”角色，点击按钮无效，请假单状态不会变化。此处有点小 Bug，据说下一版本中，不拥有相应角色时，将看不见相应按钮。

二、 请假模块结构

请假模块各文件及目录如下：

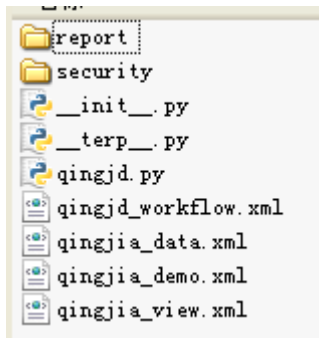


图 六-1 字段查询的过滤条件

- `__init__.py`：和普通 Python 模块中的 `__init__.py` 作用相同，是每个 OpenERP 模块必须的。
 - `__terp__.py`：OpenERP 模块特有的，详细内容见后文，是每个 OpenERP 模块必须的。
 - `qingjd.py`：请假单对象定义代码。
 - `qingjia_view.xml`：请假单视图（列表和表单视图）、动作（Action）、菜单定义文件。
 - `qingjd_workflow.xml`：请假 workflow 定义文件。
 - `qingjia_data.xml`：模块基础数据定义文件，本例中的员工和经理信息定义在该文件中。基础数据是模块正常运行必须的数据。
 - `qingjia_demo.xml`：演示数据定义文件，本例实际未使用该文件。演示数据不是模块必须的数据。
 - `security`：权限设置文件。通常包括两部分，一是定义新的权限组(group)和角色，二是定义权限组对对象(Object)的访问权限（读写建删）。
 - `report`：报表定义相关的文件。本例的“打印请假单”报表在此定义。
 - `process`：企业流程（Enterprise Process）定义相关文件，本例未使用。
- 上述各个部分，只有 `__init__.py` 和 `__terp__.py` 是必须的，其他的都不是必须的。

本例 `__terp__.py` 文件内容如下：

```
"name": "Employee Leave Example",
"author": "NewZN",
"version": "0.1",
"depends": ["base", "process", "hr"],
"description": """《OpenERP 应用和开发基础》代码模块开发示范例子。
""",
"init_xml": [],
"update_xml": [
    'security/qingjia_security.xml',
    'security/ir.model.access.csv',
    'qingjia_view.xml',
    'qingjd_workflow.xml',
    'report/rpt_qingjd.xml',
```



```
'qingjia_data.xml',
],
"demo_xml": [],
"installable": True,
```

其中：

- **name** : 模块名，任意文字。
- **depends** : 依赖模块，即安装本模块时将检查此处定义的模块，如果没安装，将自动一起安装。通常所有模块都要依赖 **base** 模块。本例工作流要依赖 **process** 模块，员工及部门经理关系用到 **hr** 模块。
- **description** : 模块描述，任意文字。
- **init_xml** : 当 Server 以参数 `- init=module` 启动时要加载的 XML 数据文件。格式是“路径+文件名”，路径是相对模块目录的路径。
- **update_xml** : 模块安装和升级时需要重新加载的 XML 文件，视图、报表、权限、工作流等的定义文件通常放在此处。通常权限定义文件放在前面，因为其它文件常引用权限定义数据。
- **demo_xml** : 演示数据文件。
- **installable** : 通常固定为 `True` 。

第二节 理解 OpenERP 的代码

OpenERP 的代码实际上是以 Python 及 XML 格式定义对象、视图、动作（Action）、菜单、工作流、报表、企业流程、权限组、角色等。这些基本概念前述章节已经介绍过了，本节主要介绍和前述章节不同之处。

一、对象定义

本例的请假单对象定义代码如下：

```
# -*- encoding: utf-8 -*-
from osv import osv, fields
class qingjia_qingjd(osv.osv):
    _name = 'qingjia.qingjd'
    _description = '请假单'
    def get_employee(self, cr, uid, context={}):
        obj = self.pool.get('hr.employee')
        ids = obj.search(cr, uid, [('user_id','=',uid)])
        res = obj.read(cr, uid, ids, ['id','name'], context)
        return res and res[0]['id'] or 0
    _columns = {
        'shenqr': fields.many2one('hr.employee', '申请人', required=True),
        'tians': fields.float('请假天数', required=True),
        'kaisrq': fields.date('开始日期', required=True),
        'shiyous': fields.text('请假事由'),
        'active': fields.boolean('有效'),
        'state': fields.selection([('draft','草稿'),('wait_prove','待批'),('proved','已批'),('rejected','被拒')], '状态',
required=True)
    }
    _defaults = {
        'shenqr': lambda self,cr,uid,context: self.get_employee(cr,uid,context),
        'active': lambda *a: 1,
```

```
'state': lambda *a: 'draft',
}
qingjia_qingjd()
```

请假单对象和之前章节中界面上定义的对象基本相同。值得注意的有：

- 1) 第一行# -*- encoding: utf-8 -*-，这行指出文件编码格式为 utf-8。当代码行中有汉字时，必须有本行，而且，文件必须以 utf-8 格式保存。通常，多数编辑器的缺省保存格式是 gb2312，要注意手工改动保存格式。
- 2) 对象名和字段名没有“必须以 x_开头”的限制。
- 3) 申请人字段，和前面章节不同的是，此处是到 hr.employee 的 many2one 关系。主要是因为权限设置时要用到主管等部门信息，这些信息在 employee 中，res.users 中没有。另外，此处设定申请人缺省值是当前登录用户对应的 employee。该对应的 employee 通过函数 get_employee 取得。
- 4) Active 字段，OpenERP 的多数对象带有该字段。该字段的作用是，如果 Active 为 False，OpenERP 将自动过滤掉本记录。这相当于“软删除”记录，即数据库中还有，但任何界面上都找不到该条记录。
- 5) State 字段，设置了该字段的缺省值为“draft”（'state': lambda *a: 'draft'）。lambda 是 Python 的语法，相当于函数的简写格式。'state': lambda *a: 'draft' 等同于下述方法定义：

```
def xfun(self, cr, uid, context):
```

```
    return 'draft'
```

```
state=self.xfun( cr, uid, context)
```

- 6) 最后一行的 qingjia_qingjd()，表示创建请假单对象，并将请假单对象放入对象池中。以后只要用语句 self.pool.get('qingjia.qingjd') 即可取得该对象。通过该对象即可对相应数据表进行增、删、查、改等数据库操作。
- 7) 对象定义的完整属性如下：

必须属性

 _name 对象名

 _columns 对象字段定义

可选属性

 _table 对象对应的表名

 _description

 _defaults

 _order

 _rec_name name 字段对应的数据库字段名

 _auto 是否自动创建数据库表

 _constraints

 _sql_constraints

 _inherit

 _inherits

上述属性的详细含义请参考官方文档。

- 8) OpenERP 支持的字段类型如下：

基础类型

 char, text

 boolean, integer, float

 date, time, datetime

 binary

复杂类型

 selection, function, related

关系类型

```

one2one
one2many
many2one
many2many

```

上述类型的详细用法请参考官方文档。

二、 视图、动作及菜单定义

视图、动作和菜单定义，参看文件 `qingjia_view.xml`。视图定义和前面章节基本相同，不同之处是，申请人和状态字段设为只读。也就是限定，状态不可任意改变，申请人只可以为自己。下面重点看看请假单审批菜单。

```

<record model="ir.actions.act_window" id="action_qingjia_qingjdprove">
  <field name="name">请假单审批</field>
  <field name="res_model">qingjia.qingjd</field>
  <field name="domain">[(('state','=','wait_prove'),('shenqr.parent_id.user_id','=','uid'))]</field>
</record>
<menuitem name="请假单审批" parent="menu_qingjia_main"
  id="menu_qingjia_qingjdprove" action="action_qingjia_qingjdprove"/>

```

值得注意的是：

1) OpenERP 的 XML 的数据定义格式。其数据定义的通用格式如下：

```

<record model="..." id="...">
  <field name="...">Value</field>
  .....
</record>

```

`<record>` 定义在数据表中增加一条记录。其中，`model` 是对象名，也等同于数据表名。`id` 是任意值，但 `id` 中的“.”有特殊含义，它是模块分界符。`id` 标志本条数据，在本模块范围内必须唯一。引用其它模块的数据时，必须用“`module.name.id`”引用。`field` 定义字段值，其“`name=`”指明字段名，`Value` 是字段值。

本例的 `<record>` 在数据表 “`ir_actions_act_window`” 中插入一条记录 (`name=`‘请假单审批’，`res_model=`‘`qingjia.qingjd`’，`domain=`...)。

2) 本 Action 通过 Domain 条件表达式 “`[(('state','=','wait_prove'),('shenqr.parent_id.user_id','=','uid'))]`”，限定本 Action 仅显示状态为“待审批”的，且申请人的主管(`shenqr.parent_id.user_id`)是当前登录用户(`uid`)的请假单。

3) `menuitem` 中的 `action="action_qingjia_qingjdprove"` 定义本菜单关联的 Action。

4) 可以指定每个 Action 的视图。每个 Action 可以同时有 “`tree, form, calendar, graph, gantt, process`” 六个视图，它们分别对应 Web 界面右上方的六个按钮。最常见的是 `tree`（列表）和 `form`（表单）视图。如果不指定 Action 的视图，系统自动采用 Action 中的对象 (`res_model`) 关联的默认视图，本例是“请假单”对象的默认列表和表单视图。

```

<record id="stock.action_picking_tree6_view1" model="ir.actions.act_window.view">
  <field eval="1" name="sequence"/>
  <field name="view_mode">tree</field>
  <field name="view_id" ref="stock.vpicktree"/>
  <field name="act_window_id" ref="stock.action_picking_tree6"/>
</record>

```

上例定义 Action “`stock.action_picking_tree6`”（库存模块中的 Action: `action_picking_tree6`）的列表视图为 “`stock.vpicktree`”（库存模块中的视图: `vpicktree`）。`view_mode` 指定视图类型，即 “`tree, form, calendar, graph, gantt, process`” 之一。`Sequence` 指定视图的显示顺序，数字越小越先显示，通常 `tree` 总是最先显示。

三、 workflows和报表定义

工作流定义文件参看 qingjd_workflow.xml, 和前面章节不同的是增加了工作流角色:

```
<field name="role_id" ref="role_depmanager" />
```

角色 "role_depmanager" 在文件 "security/qingjia_security.xml" 中定义。此处 ref 表示引用该条角色定义记录。在 qingjia_data.xml 中, 定义经理时, 下述语句赋予经理 "部门经理" 角色。

```
<field name="roles_id" eval="[(6,0,[ref('role_depmanager')])]" />
```

报表定义参看 report 目录下的相关文件。报表和前述章节完全一致, 没有改变。

四、 权限定义

文件 security/ir.model.access.csv 定义权限组对对象的访问权限。

```
"id","name","model_id:id","group_id:id","perm_read","perm_write","perm_create","perm_unlink"
"access_qingjia_qingjd_depman","qingjia_qingjd_depman","model_qingjia_qingjd","qingjia.group_depmanager",1,1,1,1
```

id 和 name 字段任意取值, model 字段格式为: model_modulename_objectname, 如请假单对象是 "model_qingjia_qingjd"。"qingjia.group_depmanager" 表示 qingjia 模块中的组 "group_depmanager", group_depmanager 是权限组 "qingjia / department manager", 在 security/qingjia_security.xml 中定义。

文件 security/qingjia_security.xml 定义组, 角色, 及组对菜单的访问权限。

第三节 服务端动作(Server Action)

Server Actions 的作用是, 当特定工作流迁移到特定状态时, 触发 Server Action 执行更多动作。如:

- 订单审批通过时, 自动在订单对应的客户中记录一些信息。
- 发票确认时, 自动 Mail 通知有关人员。
- 当一个 CRM 案例结束时, 自动弹出一个窗口要求填入总结信息。

本节通过服务端动作(Server Action)实现请假单送批时自动发 Mail 通知主管经理的要求。

一、 配置 Server Action

菜单: 管理员 → 自定义 → Action → Server Action, 新建一个 Server Action, 这个 Server Action 将在请假申请人按下请假 "送批" 按钮时发 Email 通知主管审批。本 Server Action 的各字段说明如下:

- Action Name: 任意取名。
- Object: 本 Action 要操作的对象, 即本 Action 将读取或修改该对象的数据。
- Action Type: Server Action 的类别。OpenERP 提供了多种 Server Action, 本例选用 Email, 其他类型后文再介绍。
- Sequence: 当对应同一动作, 定义了多个 Server Action 时, 本 Action 的执行顺序。数字越小越先执行。
- Condition: 本 Action 是否执行的检查条件。

这几个字段是所有类型的 Server Action 的共同字段, 下面看看 Email Server Action 的特有字段。

- Email Address: 自然是收件人的Mail地址。可以直接写地址, 如 abc@homerp.com。也可以引用对象中

的数据，如下图所示是请假单申请人的主管的工作Mail（object.shenqr.parent_id.work_email）。注意，此处的对象引用不要用“[[]]”括起来。

- Subject: 其意自明，不用介绍。可用[[]]引用对象数据。
- Message: Mail 正文，可用[[]]引用对象数据，如下例，Mail 开头是请假申请人的主管的名字（object.shenqr.parent_id.name）。

图 六-2 字段查询的过滤条件

二、 配置 OpenERP Mail

要使 OpenERP 发送 Mail，需要在配置文件中配置下述项目：

```
smtp_port = 25
smtp_server = 202.120.88.78
smtp_user = xxx@cc.ecnu.edu.cn
smtp_password = yyy
email_from = xxx@cc.ecnu.edu.cn
```

- smtp_port: 通常是 25，这是 SMTP 协议的默认端口。
- smtp_server: 你的 Mail 服务器。如果不知道你的 Mail 服务 IP，可用 nslookup 命令查找，如下例查找“cc.ecnu.edu.cn”对应的 Mail 服务器。

```
[root@localhost site-packages]# nslookup -q=mx cc.ecnu.edu.cn
Server:          202.121.80.10
Address:         202.121.80.10#53
```

Non-authoritative answer:

```
cc.ecnu.edu.cn  mail exchanger = 10 webmail.ecnu.edu.cn.
cc.ecnu.edu.cn  mail exchanger = 5 mailscan.ecnu.edu.cn.
```

Authoritative answers can be found from:

```
cc.ecnu.edu.cn  nameserver = liwa.ecnu.edu.cn.
```

- `smtp_user`: 登陆 mail 服务器的用户名，通常是一个有效的 Mail 地址。
- `smtp_password`: 登陆 Mail 服务器的密码。OpenERP 将用这里配置好的 Mail 用户名和密码发送 Mail。
- `email_from`: 显示在系统发出的 Mail 的 From 栏的 Mail 地址。通常和 `smtp_user` 一样。

三、 测试通知 Mail

开始测试之前，还要先修改下 OpenERP 当前版本(本书用的是 Windows 版的 5.06 All In One 版)的几个 Bug。一是不知何故，在 OpenERP 的安装目录下的 Server 下的 library.zip 中，email\mime 下没有任何东西。（也许是 Tiny 公司故意制造某些麻烦，好让用户找他们培训，呵呵）。修正方法是，用 Winrar 或其他工具打开 library.zip，删除其中的 email 目录。再从 OpenERP 安装目录下的 Web\python25\Lib 下的 email 目录拖进 library.zip。Web\python25\Lib 下的 email 是未编译的源码，它下面的 mime 很齐全，原样拖进去即可。注意，不要用 WinRAR 解压开 library.zip，再替换，再压缩成 library.zip。这样的话，将导致 OpenERP Server 无法启动（原因尚不明）。

第二个问题是，目前版本的 OpenERP，对于中文 Email（Subject 或正文中有汉字的 Mail）会报错。修改方法是，Server\addons\base\ir\ir_actions.py (版本 5.06)文件中的下述几行：

```
Line 484: return str(result) → return tools.ustr(result)
Line 549: str(action.subject) → tools.ustr(action.subject)
Line 550: str(action.message) → tools.ustr(action.message)
```

原文中用 str 编码时不能处理 utf-8 的文字，必须改为 ustr 编码。修改完成后重启 Server，以使修改生效。而后，进入菜单：人力资源 → 员工 → 所有员工，找到“经理 1”，点击编辑，在工作 mail (Work Email) 中添加经理的 Mail 地址。再进入菜单：管理员 → 自定义 → 工作流，选择“请假申请工作流”，打开，编辑 Activity “wait_prove”，在该 Activity 中增加前面配置的 Server Action：请假申请 Mail 通知。如下图所示：

Activity

Name:	wait_prove		
Workflow:	请假申请工作流	Kind:	Function
Server Action:	请假申请Mail通知		
Python Action:	write({'state':'wait_prove'})		

图 六-3 字段查询的过滤条件

这个配置表示，当用户按下请假工作流的“送批”按钮时，迁移到 Activity: “wait_prove”，触发 Server

Action: 请假申请 Mail 通知。该 Action 发送 Mail 给主管。

登出系统，以员工 1 身份(yg01)登陆，创建一张请假单，送批，注意，送批动作有点慢。主管将收到通知 Mail，如下：



图 六-4 字段查询的过滤条件

如果发送成功，在系统日志中会出现如下日志信息：

INFO:email:Email successfully send to : xyz@gmail.com

如果发送失败，日志信息如下：

ERROR:email:[11]: Last revision No. & ID : Failed to send email to : xyz@gmail.com

四、 更多 Server Action

1. Dummy: A dummy stage does nothing. (God knows why its there..)
2. Client Action: You Can select a form to Open, a wizard to run or a report to launch. Select the report from the list.
3. Iteration: Based on a python loop expression, you can iterate server actions, eg: When a stock inward move occurs and you confirm it, you want each line item to be historised.. you can loop on expression `object.move_lines` and create another server action which is referred to do the historising job.
4. Python Code: you can execute a multiline python code. the returned value is the value of the variable `action = { }`. This makes sense only if you want to pop a specific window(form) specific to the context. IMHO you wont need a return value. Note: The code is executed using the `exec` function of python, which is run in the dictionary namespace with variables: `object,time,cr,uid,ids`
5. Trigger: Any transition of the workflow can be triggered using this. The options you need to set are. Workflow Model: The target object on which you want to trigger the workflow. 'Trigger on' field should have the ID of the target model record. Eg have the ID of invoice if you want to trigger a change in invoice. 'Trigger Name' is the signal you have to use to initiate the transition. The drop down lists all possible triggers. Note:Trigger Name list shows all possible transitions from ther models also, so ensure you select the right trigger. Models are given in bracket.
6. Email: Set an email address, Subject & message. You need to configure the Open ERP building smtpserver for this. Power email, a generic email architecture for Open ERP can be used for emailing as it offers a lot more features and automated emails (no messing with server actions required). Its available at: <http://launchpad.net/poweremail> and downloads at <http://launchpad.net/poweremail/+download>
7. SMS: Choose mobile no. eg: `[[object.shipping_address.mobile]]` and the subject.
8. Create Object: This is used to create a new record in any model, when the server action is triggered. The historise feature discussed can be implemented using this. Field mappings are used to give value for fields in record. Lets try an example of creating events for sale order.

1. Destination: Target field for which you are assigning value: eg. `partner_id`

2. Type: Formula is evaluated using eval and value is directly applied

3. Value/formula: if value just enter it. For partner id the formula will be `object.partner_id` (refers to `partner_id` in sale order). Refer to screen shots for more, note that step two has to be completed for actions to run. These screenshots are here only to help understanding. Also note that there is no need of `[[[]]]` in formulas. fromulas are directly evaluated.

4. Create id: you can choose the field to which the ID of the newly created record has to be saved. In history this is not necessary, however if you have a m2o relationship and want to keep the ID then this is useful

9. Write Object: Similar to create object except that it is used to modify an existing record identified by the `write_id`. If you are updating a record you previously created in the create, then the same field for create id can be used for write id.

10. Multi actions: as explained in the context of sequence, you can create multiple server actions and call them one by one in the order of their sequences.

第七章 超强的继承功能

TODO

第八章 基于 XML-RPC 的集成

TODO