ODOO LEARNING DOCUMENT

Important:

admin@nitrawork.com

artin.nitrawork

Many to One : Multiple entities to choose from but only 1 can be selected(Her zaman id ile seçim yapılır)

Many to Many: Multiple entities to choose from and multiple of them can be selected(1 version of them)

One to Many: 1 entity to choose from but it can have many versions

!!! Russian text causes semantic errors if used in tuples e.g. (“cупер”) !!!

!!! If you are modifying a field that is fetched by the code, do not set it empty via the code, consider the empty case !!!

Navigation:

Dev : Developer panel button top-right

----------------------------

Technical -> Logging

Technical -> Automated Actions

Technical -> Actions -> Server Actions

Dev -> Edit Action

Dev -> View Fields

Contacts.Dev -> Edit view : Form

Dev -> View Metadata

Invoicing -> Payment Terms

Users & Companies -> Users

############################

GENERAL

How to copy, restore, move databases:

1. !!! <http://localhost:8069/web/database/manager>!!!
2. If you have external apps installed in a database that you are installing, you have to install it again via placing inside the addons folder otherwise you will get INTERNAL SERVER ERROR

How to find id of specific fields:

* App -> Configuration –> find the field -> select or create one -> Dev -> View Metadata -> ID

How to trigger a warning :

* raise UserError("Lütfen emailde @ sembolünü kullanın!!")

How to install external apps into Odoo :

1. Put the addon’s folder into Odoo’s addon folder.
2. Odoo -> apps -> Update Apps List
3. Restart Odoo server

How to import excel files:

* Contacts -> Favourites -> Import records

Dashboard Ninja installation error( Clicking My Dashboard launches this error):

* Restart Odoo server

Good theme :

* Vista backend ( Good with Dashboard Ninja )

Odoo Server Error with empty log:

* Generally happens if you have wrong type conversions in the python code or situations that result in empty fields

USERS & COMPANIES

User types:

1. Internal User : Has all rights
2. Portal : Able to login but can’t control many things
3. Public : Can’t login, only able to view 00

How to add new users into Odoo :

* Users & Companies -> Users -> Create

How to change language of an user:

* Users & Companies -> Users -> user -> Edit -> Preferences -> Language

How to add custom Access Rights to a user:

1. Users & Companies -> Groups -> Go into one of the most used ones( eg. Sales ) -> Action -> Duplicate
2. Go into duplicated -> Tweak to your needs
3. Go into the user you want -> Search for the ‘Application’ you have set the rights for -> Tick the box

TRANSLATIONS

How to change language Odoo wise:

* Translations -> Languages -> select the language -> Activate or Disable

How to add more translations:

1. Translations -> Translated Terms -> Create or Edit(all instances)
2. Translations -> Generate Missing Terms -> select the language -> Update

ODOO BACK-END / TECHNICAL

!!! [Every](http://localhost:8069/web/database/manager) instruction starts with Technical ->!!!

Automated actions: Required when we need to automate an action e.g. We want to show a warning if a user is created w/out a mobile number.

Scheduled actions: Required when we need timed actions e.g. we want to update sale dates every 1 minute

Server actions: Required when we need to set a button and trigger a script from that button e.g. we want to analyze all sales and display the total revenue

ACTIONS

How to add server actions:

* Actions -> Server Actions -> Create

How to add a button :

1. Actions -> Server Actions -> Name it and write your code -> Create contextual action.
2. Go to your app -> List view -> select a record-> Action -> your button is there.
3. Now you can press it and it will trigger your action

AUTOMATION

How to add automated actions:

* Automation -> Automated Actions -> Create

How to add scheduled actions:

* Automation -> Scheduled Actions -> Create

DISCUSS

How to check system-wide messages :

* Discuss -> Messages

How to set activities:

1. Discuss -> Activity Types -> Create
2. e.g. Sale app -> Next Activity -> click on clock symbol -> Schedule an activity -> select your activity

How to block e-mails or phones from the site from logging in:

* For emails: Discuss -> Email Blacklist
* For mobiles: Phone / SMS -> Phone Blacklist

How to configure user settings for communication:

* Discuss -> User Settings

How to configure video call settings:

* Discuss -> RTC sessions

EMAILS

How to send periodic result e-mails of the odoo:

* Emails -> Digest Emails

How to send e-mails:

1. Creating the Outgoing Mail Server: Technical -> Email -> Outgoing Mail Servers -> Create
2. For @gmail.com configuration:

* SMTP Server: smtp.gmail.com
* SMTP Port: 465
* Connection Security: SSL/TLS
* Username: your test email
* Password: email’s password( create app password after setting 2 step authentication from gmail’s security settings )

1. Test Connection and check if it’s successful
2. Creating the Email Template: Technical -> Email -> Email Templates -> Create (copy an existing one for easier time)
3. Edit the content.
4. Email Configuration ( for Contact module ):

* From : use {{ user.email\_formatted }} -> user is the user of odoo
* To (Emails) : {{ object.email\_formatted }}

1. Creating a trigger: Technical -> Automated Actions -> Create

* Action to Do: Send Email
* Email Template: Select the template that you created

1. Trigger your action and go to Technical -> Email -> Emails
2. Check if you have your email ready, send if necessary

USER INTERFACE

How to add new menu items or change them:

* User Interface -> Menu Items

How to revert changes to a view system-wise:

* User Interface -> Views -> Action -> Compare/Reset -> select what you want

How to add custom filters: This operation can be done via Dev -> Edit: ControlPanelView using XML code

1. User Interface -> User-defined Filters -> Create
2. Domain -> customize it

* Context : If you want to group by a specific attribute. e.g. {'group\_by':'name'}

1. Go to your configured model -> Favorites -> select your filter

How to change Odoo’s theme:

* User Interface -> Change Backend Theme

How to configure Odoo tours:

* User Interface -> Tours
* Dev -> Start Tour / Disable Tour

DATABASE STRUCTURE

All assets: Database Structure -> Assets

All models: Database Structure -> Models

All fields: Database Structure -> Fields

All logs: Database Structure -> Logging

All ManyToMany Relations: Database Structure -> ManyToMany Relations

Model constraints: Database Structure -> Model Constraints

How to change digit accuracy of numeric values system-wise:

* Database Structure -> Decimal Accuracy

How to see all images and css/js files:

* Database Structure -> Attachments

REPORTING

How to change paper formats system-wise:

* Reporting -> Paper Format

How to add Print button and option to download PDF of a given page:

1. Actions -> Reports -> Select/Create
2. Configure fields and press QWeb views -> write your PDF code
3. Go to configured model and Print button should be there

CODE

How to send data to Odoo from Python code( e.g create contact ):

1. Have this code first, to login to Odoo with your credentials:
   1. import requests
   2. import json
   3. url = 'http://localhost:8069/web/session/authenticate'
   4. headers = {
   5. "Connection": "keep-alive",
   6. "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5",
   7. "Content-Type": "application/json"
   8. }
   9. myobj = {
   10. "jsonrpc": "2.0",
   11. "params": {
   12. "login": "kilicarslan.business@gmail.com",
   13. "password": "0987612345",
   14. "db": "modem"
   15. }
   16. }
   17. x = requests.post(url, json = myobj, headers=headers)
   18. #print the response text (the content of the requested file):
   19. #return str(x.content)
   20. #response = x.json()
   21. #return str(response['jsonrpc'])
   22. #aşağıdaki işlemle önce json parse edildi sonra 0-52 ye kadar substring yapıldı
   23. global cookie
   24. cookie = ((x.headers)['Set-Cookie'])[0:51]
   25. print(cookie)
2. Go to your app and enter a name, just before you press create -> right click -> inspect ->
3. Network -> press create -> you’ll see “create” event below -> right click -> copy -> Copy as cURL( bash )
4. Open Postman app -> Import -> Raw text -> paste your cURL -> Continue -> import
5. Copy the created url -> go to code :
   1. url = 'http://localhost:8069/web/dataset/call\_kw/res.partner/create'
   2. headers = {
   3. "Connection": "keep-alive",
   4. "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5",
   5. "Content-Type": "application/json",
   6. "Cookie": cookie
   7. }
   8. myobj = {"id":33,"jsonrpc":"2.0","method":"call","params":{"args":[{"partner\_gid":0,"additional\_info":False,"image\_1920":False,"\_\_last\_update":False,"is\_company":False,"active":True,"company\_type":"person","name":"proto5","parent\_id":False,"company\_name":False,"type":"contact","street":False,"street2":False,"city":False,"state\_id":False,"zip":False,"country\_id":False,"vat":False,
   9. "x\_uptime":"5h 32m 12s",
   10. "x\_wireless\_status":"enable",
   11. "x\_channel":"auto",
   12. "x\_mac":"1c:18:4a:47:63:80",
   13. "x\_ip":"192.168.5.1",
   14. "x\_subnet":"255.255.255.0",
   15. "x\_dhcp":False,
   16. "x\_ssid":False,
   17. "x\_enable\_wireless":True,
   18. "x\_enable\_ssid":True,
   19. "x\_manual\_time":False,
   20. "x\_new\_password":False,
   21. "x\_wan\_status":False,
   22. "x\_lan\_info":False,
   24. "phone":False,"mobile":False,"user\_ids":[],"email":False,"message\_follower\_ids":[],"activity\_ids":[],"message\_ids":[]}],"model":"res.partner","method":"create","kwargs":{"context":{"lang":"tr\_TR","tz":"Europe/Istanbul","uid":2,"allowed\_company\_ids":[1],"params":{"id":11,"cids":1,"menu\_id":96,"action":122,"model":"res.partner","view\_type":"form"},"default\_is\_company":True}}}}
   26. x = requests.post(url, json = myobj, headers=headers)
   27. #print the response text (the content of the requested file):
   28. #return str(x.content)
   29. #response = x.json()
   30. #return str(response['jsonrpc'])
   31. #aşağıdaki işlemle önce json parse edildi sonra 0-52 ye kadar substring yapıldı
   32. last\_result = json.loads((x.content))
   33. print(str(last\_result))
6. Change necessary fields and run the program.
7. You will see your newly created contact in Odoo.

How to send data to Odoo from Python code, the better and faster way( e.g create modems ):

1. Example dictionary with a list inside of it. Inside this list will be multiple dictionaries that’ll be fed to it. These dictionaries will be individual records of a model e.g. each contact in the Contact module:
   1. modem\_read\_result\_list = {"modems":[]}
   3. while not read\_queue.empty():
   4. modem\_read\_result\_list["modems"].append(read\_queue.get())
   6. send\_datato\_odoo(modem\_read\_result\_list)
2. Send\_datato\_odoo function:
   1. def send\_datato\_odoo(modem\_data: dict):
   2. url = 'http://localhost:8069/create/modems\_from\_data' *# need to check this for multiple databases position*
   3. headers = {
   4. "Connection": "keep-alive",
   5. "User-Agent": "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like Gecko) Chrome/19.0.1084.52 Safari/536.5",
   6. "Content-Type": "application/json",
   7. "Cookie": cookie
   8. }
   10. requests.post(url, json = modem\_data, headers=headers)
3. Now go to your model’s folder -> controllers -> controller.py. Put this function under your model class:
   1. class modemProfileReq(http.Controller):
   2. @http.route(['/create/modems\_from\_data'], type="json", auth="public", methods=["POST"], cors='\*', csrf=False)
   3. def modem\_data\_send(self):
   4. data = json.loads(request.httprequest.data)
   5. for modem in data["modems"]:
   6. http.request.env['modem.profile'].sudo().create({
   7. 'x\_uptime': modem['x\_uptime'],
   8. 'x\_wireless\_status': modem['x\_wireless\_status'],
   9. 'x\_channel': modem['x\_channel'],
   10. 'x\_mac': modem['x\_mac'],
   11. 'x\_device\_info': modem['x\_device\_info'],
   12. 'x\_ip': modem['x\_ip'],
   13. 'x\_subnet': modem['x\_subnet'],
   14. 'x\_dhcp': modem['x\_dhcp'],
   15. 'x\_enable\_wireless': modem['x\_enable\_wireless'],
   16. 'x\_enable\_ssid1': modem['x\_enable\_ssid1'],
   17. 'x\_enable\_ssid2': modem['x\_enable\_ssid2'],
   18. 'x\_enable\_ssid3': modem['x\_enable\_ssid3'],
   19. 'x\_enable\_ssid4': modem['x\_enable\_ssid4'],
   20. 'x\_manual\_time': modem['x\_manual\_time'],
   21. 'x\_new\_password': modem['x\_new\_password'],
   22. 'x\_hotel\_name': modem['x\_hotel\_name'],
   23. 'x\_update\_date': modem['x\_update\_date']
   24. })
4. Now when send\_datato\_odoo is called with that dict of list of dicts object, it will send this request to the url specified and the controller will grab it and then unpack its values, create new records with its data.

READ

search():

List\_Olga = env['res.partner'].search([('name','=', 'Olga')])

List\_Turkey = env['res.partner'].search([('country\_id', '=', 224)]

Contacts – Country changer according to phone number  
for rec in records:

if "+90" in rec.mobile:

rec["country\_id"] = 224

Contacts - Number converter:

for rec in records:

if "Olga" in rec.name:

rec["mobile"] = "+790"

Contacts - Bad words warning:

for rec in records:

if "şerefsiz" in rec.comment or "salak" in rec.comment:

raise osv.except\_osv(\_('warning'), \_('Lütfen uygun kelimeler kullanın!!'))

Contacts – Logging sales amount when a contact is created:

List\_Record = env['sale.order'].search([])

log(str(len(List\_Record)), level='info')

Sales app - Combining Quotation date with a custom field named “x\_sales\_detail”:

for rec in records:

rec["x\_sales\_detail"] = "Camera Was sold in " + rec.date\_order.strftime('%m/%d/%Y')

Sales app - Showing in logs if the sale date has passed or not:

for rec in records:

if rec.date\_order < datetime.datetime.now():

log("Satışın tarihi geçti", level='info')

else:

log("Satışın tarihi gelmedi", level='info')

Sales app – Show in logs if the person that has the sale is from Turkey or Not:

for rec in records:

if rec.partner\_id.country\_id.id == 224:

log("Bu kişi Türktür", level='info')

else:

log("Bu kişi Türk değildir", level='info')

CREATE

MANY TO ONE

Sales app – Create a contact when there is a sale created:

contact\_title = env['res.partner.title'].search([('name', '=', 'Mister')], limit=1)

company\_name = env['res.partner'].search([('name', '=', 'Ali Özer')], limit=1)

company\_name\_filtered\_by\_mobile = env['res.partner'].search([('phone', '=', '0555')], limit=1)

contact = env['res.partner'].create({

'name': 'newcustomer',

'parent\_id': company\_name\_filtered\_by\_mobile.id,

'company\_type': 'person',

'function': 'Employee',

'phone': '',

'mobile': '+901232321',

'email': 'abc@mail.com',

'website': '',

'title': contact\_title.id,

'category\_id': False

})

Sales app – Create a sale with 15 days payment term and Ayşe as the customer name:

sale\_customer = env['res.partner'].search([('name', '=', 'Ayşe')], limit=1)

payment\_term = env['account.payment.term'].search([('name', '=', "15 Days" )], limit=1)

sale = env['sale.order'].create({

'partner\_id': sale\_customer.id,

'validity\_date': False,

'date\_order': datetime.datetime.now(),

'payment\_term\_id': payment\_term.id

})

WRITE

Sales app – Overwrite a sale and change its name and add an expiration date when a contact is created:

sale\_to\_modify = env['sale.order'].search([('partner\_id', '=', 'Ayşe')], limit=1)

name\_to\_paste = env['res.partner'].search([('name', '=', 'Ali Özer')], limit=1)

sale\_to\_modify.write ({

'partner\_id': name\_to\_paste.id,

'validity\_date': datetime.datetime.now(),

'date\_order': datetime.datetime.now(),

})

MANY TO MANY

Contacts app – Write a many2many field “tags” 2 different tags when the contact is updated:

contacts = env['res.partner'].create({

'name': 'newcustomer',

'parent\_id': False,

'company\_type': 'person',

'function': 'Employee',

'phone': '',

'mobile': '+901232321',

'email': 'abc@mail.com',

'website': '',

'title': False,

'category\_id': False

})

contacts.write({

'category\_id' : [(6, 0, [1, 2])]

})

ONE TO MANY

Sales app – Create sale and write two sales on the sale object :

customer\_name = env['res.partner'].search([('name', '=', 'Ali Özer')], limit=1)

product = env['product.template'].search([('name', '=', 'Kamera 1')], limit=1)

product2 = env['product.template'].search([('name', '=', 'Kamera 2')], limit=1)

sale = env['sale.order'].create({

'partner\_id': customer\_name.id,

'date\_order': datetime.datetime.now()

})

sale.write({

'order\_line' : [(0, 0, {'product\_id': product.id, 'name': product.name, 'product\_uom\_qty': 5, 'price\_unit': 100})]

})

sale.write({

'order\_line' : [(0, 0, {'product\_id': product2.id, 'name': product2.name, 'product\_uom\_qty': 2, 'price\_unit': 70})]

})

DELETE

Important way to delete:

Automated action -> Edit Domain -> Enter filters and copy the code, paste into the filter brackets.

1. env['res.partner'].search([]).unlink()
2. After copying the filters -> ["&",["country\_id.id","=",224],["name","ilike","Ali"]]
3. env['res.partner'].search(["&",["country\_id.id","=",224],["name","ilike","Ali"]]).unlink()

unlink():

env['res.partner'].search([('name', '=', 'Ayşe')]).unlink()

PARSER

Split the email via @ and log the name and domain seperately.

if "@" in record.email:

parsed = record.email.split("@")

str\_to\_pass = "Name: " + parsed[0] + " Domain: " + parsed[1]

else:

raise UserError("Lütfen emailde @ sembolünü kullanın!!")

log(str\_to\_pass, level = "info")

Take the middle part of the given name and log it(hardcoded – not good)

name = record.name[6:15]

log(name, level = "info")

Scheduled actions:

Fetch all sale records and update the order date with current time:

sales = env['sale.order'].search([])

for sale in sales:

sale["date\_order"] = datetime.datetime.now()

Take the middle part of the given name and log it, filtered by kcl\_ in start and pcl\_ in end – Defensive programming:

if "kcl\_" in record.name and "pcl\_" in record.name:

if record.name.index("kcl\_") == 0:

prefix = record.name.index("kcl\_") + 4

postfix = record.name.index("pcl\_")

name\_to\_fetch = record.name[prefix:postfix]

if len(name\_to\_fetch) > 0:

log(name\_to\_fetch, level = "info")

else:

raise UserError("Boş isim girdiniz!!")

else:

raise UserError("Lütfen kcl\_ ve pcl\_ sıralamasını karıştırmayın!!")

else:

raise UserError("Lütfen kcl\_ ve pcl\_ yi unutmayın!!")

ODOO FRONT-END

Kanban, tree(list), form -> 3 views

How to make a field invisible:

* <field name="type" attrs="{'invisible': [('name','=', ‘AAA’)]”} />

How to display a field to the right: Left, right, left, right alternating between each <group>

<group>

</group>

<group>

<field name="x\_teachers"/>

</group>

How to make a many2one field whose values are selected from a specific field:

Filtered via function = Öğretmen:

<field name="x\_teachers" options="{'no\_create': True}" domain="[('function','=','Öğretmen')]"/>

How to make it so that only a specific group can see the field:

Filtered via groups="base.group\_system": -> Admin

<field name="x\_teachers" groups="base.group\_system" options="{'no\_create': True}"/>

How to make it so that only a specific group can see the field:

<span>

İyi yorum oranı

</span>

<li class="o\_text\_overflow">

<field name="x\_score\_good"/>

</li>

How to add a button(<button>): “name” is the function to call.

<div class="row">

<div class="col-xs-12 col-xs-offset-0 col-sm-6 col-sm-offset-3 col-lg-12 col-lg-offset-0">

<button name="create\_company" type="object" class="btn btn-outline-success btn-block btn-lg" >İyi yorum oranı</button>

</div>

</div>

How to fetch data from backend (e.g. email):

* <t t-out=”object.name or ‘’”>

How to customize control panel:

* Dev -> Edit: ControlPanelView

WEBSITE FRONT-END

How to record forms into the Back-end:

1. Install CRM app
2. Odoo -> Settings -> CRM -> Tick “Leads”
3. Go into your website form and select it in the edit view
4. Form -> Action -> Create an Opportunity
5. Now whenever the form is submitted, it will appear under Odoo -> CRM -> Leads

How to change button color:

* Select the button -> Inline text -> Style -> Custom

How to translate a page:

1. Bottom of the page -> Change language
2. At the top left - > Translate -> wait and start changing the fields