**Лабораторная работа № 6.**

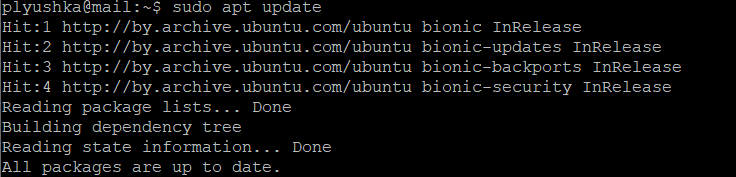
**Инфраструктура Docker.**

**Цель:** освоить навыки развертывания контейнера из образа и его настройки в среде docker.

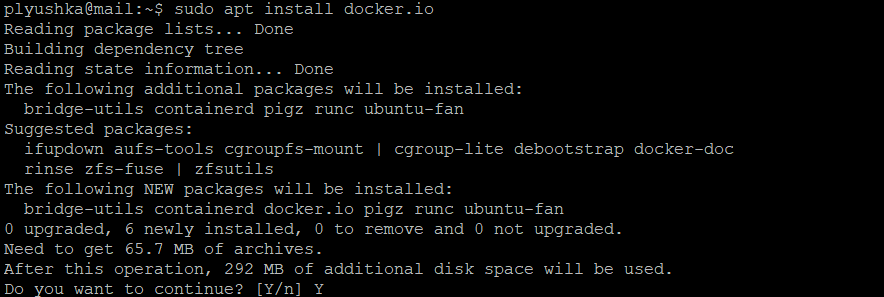
**Задачи:**

1. **на виртуальной машине через утилиту apt установить клиентский пакет docker и серверный пакет dockerd;**

Обновляем списки пакетов: sudo apt update



Установливаем клиентский пакет Docker и серверный пакет (dockerd входит в него): sudo apt install docker.io (или docker-ce, если используете официальные репозитории Docker).



Проверяем установку: docker --version



Запускаем и добавляем службу Docker в автозагрузку:

sudo systemctl start docker

sudo systemctl enable docker



Добавляем пользователя в группу docker, чтобы выполнять команды без sudo (после этого выходим из системы и заходим снова):

sudo usermod -aG docker $USER



1. при использовании университетской сети сконфигурировать dockerd на использование прокси-сервера 172.16.0.101:3128;

sudo mkdir -p /etc/systemd/system/docker.service.d:

Создает директорию /etc/systemd/system/docker.service.d.

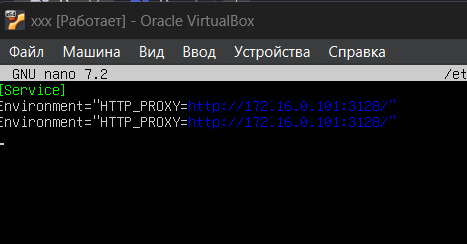
Эта директория используется для дополнительных (переопределяющих) настроек системного сервиса Docker, управляемого systemd. Опция -p создает родительские директории, если их нет.

sudo nano /etc/systemd/system/docker.service.d/http-proxy.conf:

Открывает (или создает) файл http-proxy.conf в этой директории с помощью текстового редактора nano.

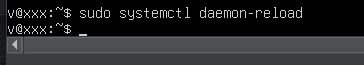
Цель — добавить или изменить конфигурацию Docker, в данном случае (http-proxy.conf), для настройки HTTP/HTTPS прокси, который Docker будет использовать (например, для скачивания образов).

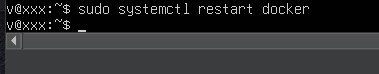
****

****

sudo systemctl daemon-reload: Заставляет систему перечитать измененные файлы конфигурации служб.

sudo systemctl restart docker: Перезапускает службу Docker, чтобы она начала работать с примененными новыми настройками.

****

****

1. **выполнить установку ERP-системы odoo с помощью docker и инструкций на сайте hub.docker.com, согласиться на внесение в базу данных демонстрационных данных;**

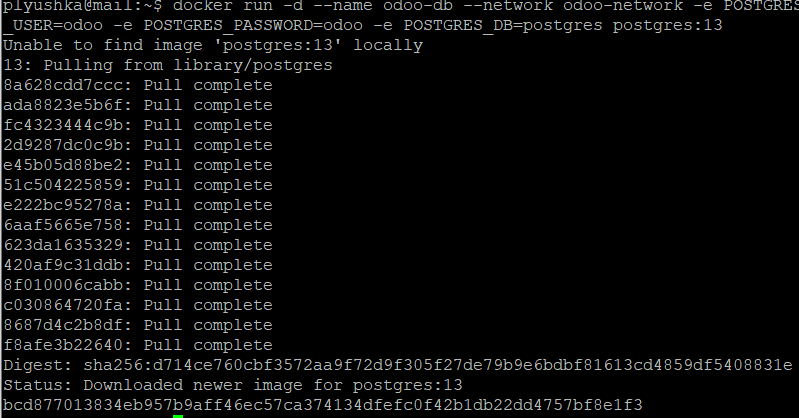
Мы будем использовать официальный образ Odoo и отдельный контейнер для базы данных PostgreSQL. Часто это проще сделать с docker-compose, но можно и через docker run.

Сначала создаём сеть: docker network create odoo-network



Запустите контейнер PostgreSQL:

docker run -d --name odoo-db --network odoo-network -e POSTGRES\_USER=odoo -e POSTGRES\_PASSWORD=odoo -e POSTGRES\_DB=postgres postgres:13



Создаём том (volume) для пользовательских модулей Odoo:

docker volume create odoo-extra-addons



Запускаем контейнер Odoo (используем Odoo 13):

docker run -d -p 8069:8069 --name odoo --network odoo-network \

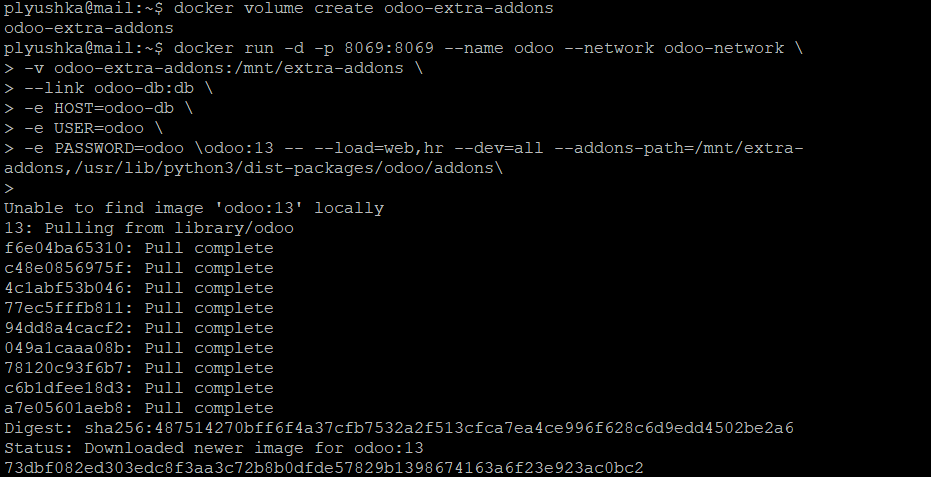
-v odoo-extra-addons:/mnt/extra-addons \

--link odoo-db:db \

-e HOST=odoo-db \

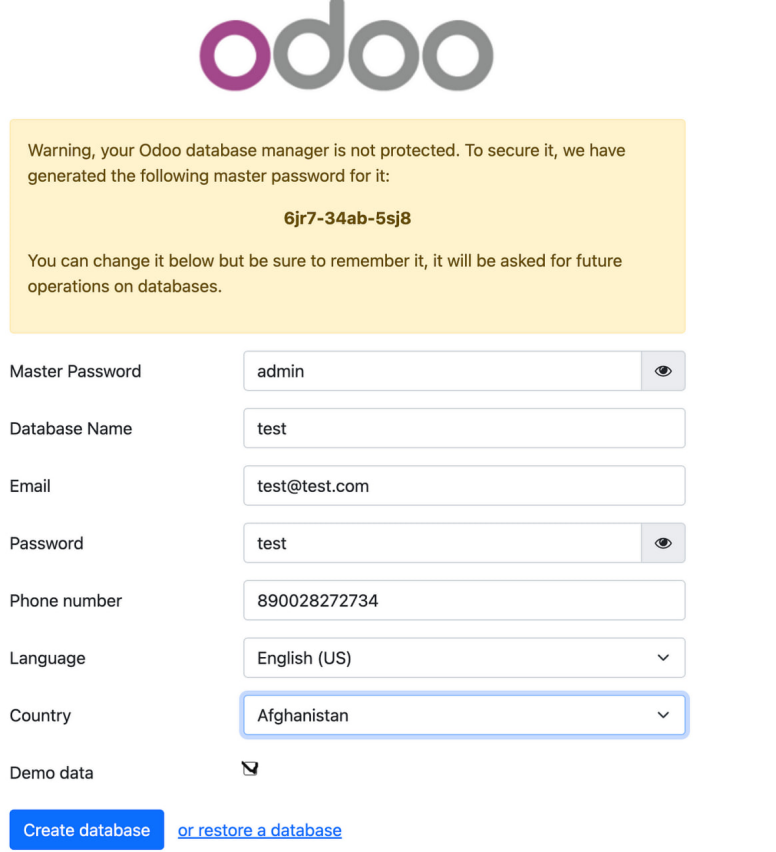
-e USER=odoo \

-e PASSWORD=odoo \odoo:13 -- --load=web,hr --dev=all --addons-path=/mnt/extra-addons,/usr/lib/python3/dist-packages/odoo/addons



Откроем Odoo в браузере: http://196.168.1.101:8069

Мы видим экран настройки базы данных Odoo. Создаём новую базу данных. Важно: отмечаем галочкой «Загрузить демонстрационные данные» (Load demonstration data). Заполняем пароль администратора, email и т.д.



1. **сконфигурировать ERP-систему odoo на использование SMTP-сервера postfix, входящего в состав пакета iRedMail, для отправки e-mail;**

Входим в админ-панель Odoo.

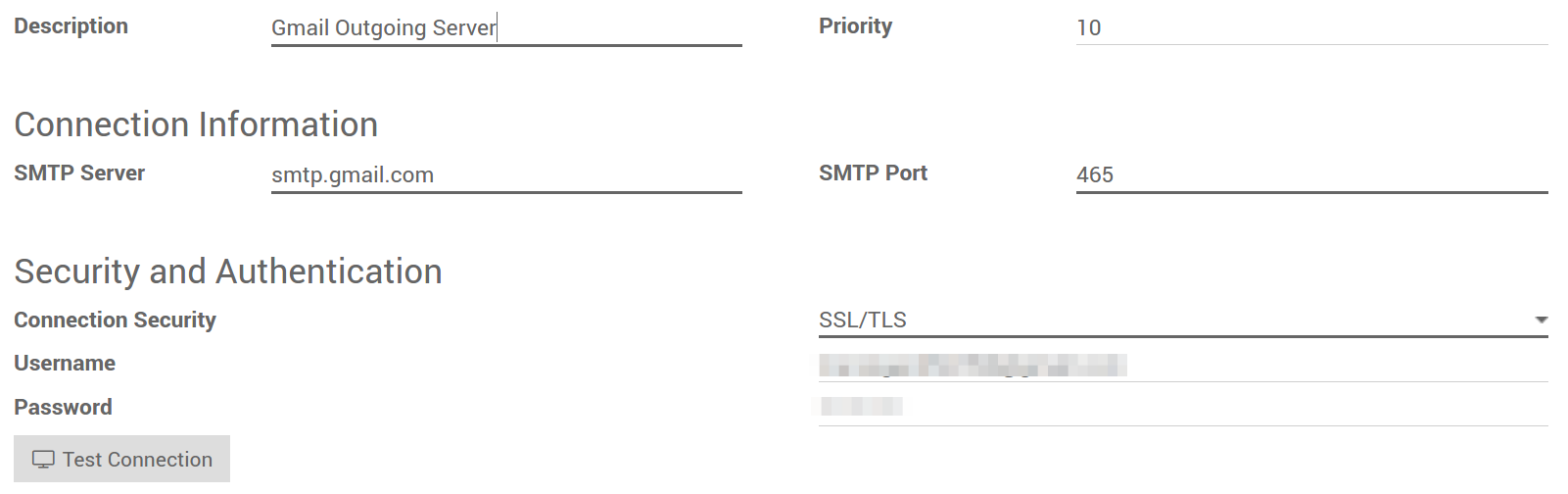
Переходим в Настройки > Технические > Электронная почта > SMTP-серверы.

Добавляем новый SMTP-сервер с параметрами:

SMTP-сервер: postfix

Порт: 465

Логин и Пароль: если требуется аутентификация, указываем данные от почтового ящика.



1. **сконфигурировать SMTP-сервер postfix на прием писем из docker-контейнера с ERP-системой odoo;**

Нам нужно разрешить IP-адресу контейнера Odoo пересылать почту через Postfix.

Находим IP-адрес контейнера Odoo в сети odoo-network: docker inspect odoo | grep IPAddress



Отредактируем конфигурационный файл Postfix (обычно /etc/postfix/main.cf на почтовом сервере).

Найдём параметр mynetworks. Добавляем сеть Docker или конкретный IP контейнера в список доверенных сетей:

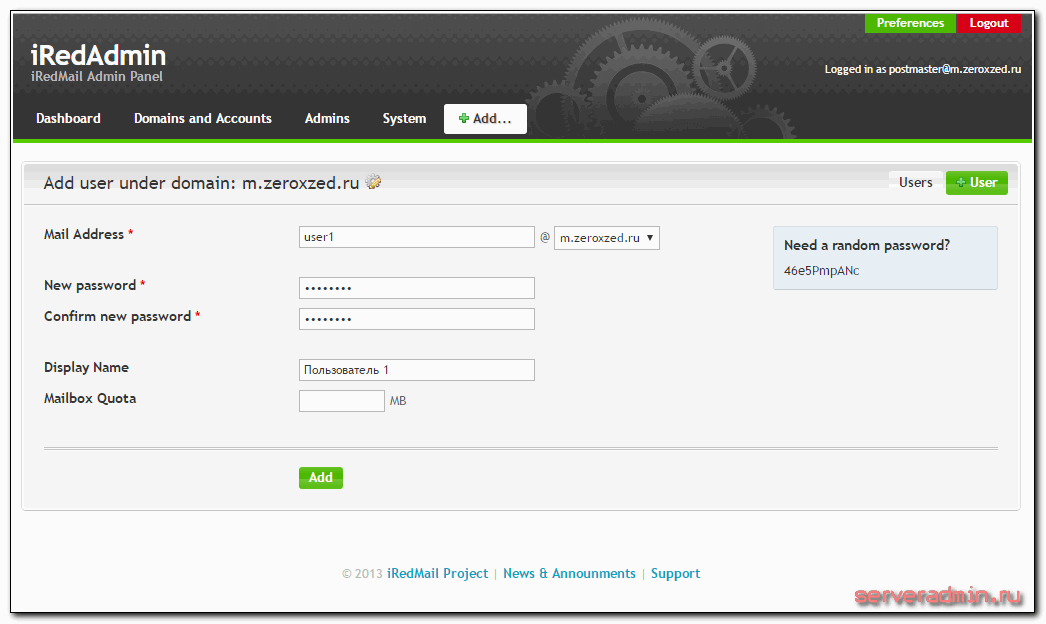
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 196.168.1.101/24 172.18.0.0/16

Добавление 172.18.0.0/16 (предполагая, что сеть Docker в этом диапазоне) доверяет всей сети Docker.

Перезагружаем конфигурацию Postfix: sudo systemctl reload postfix



1. **создать почтовый ящик на почтовом сервере с помощью панели iRedAdmin;**

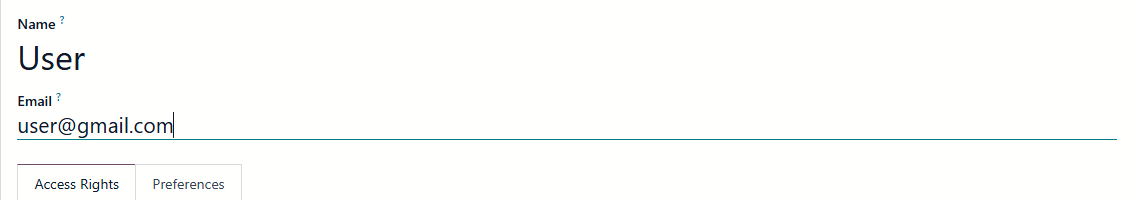


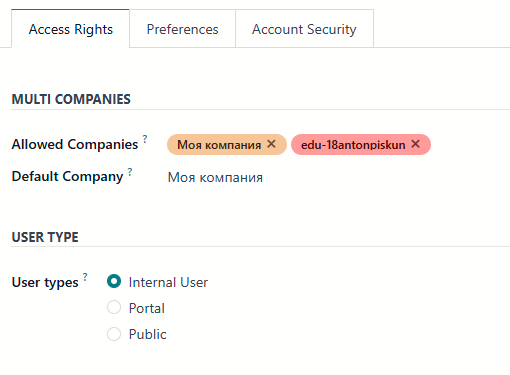
1. **создать пользователя в ERP-системе odoo, назначить ему права на использование только бухгалтерских компонентов, выслать подтверждение на созданный почтовый ящик;**

В Odoo переходим в Настройки -> Пользователи и Компании -> Пользователи.

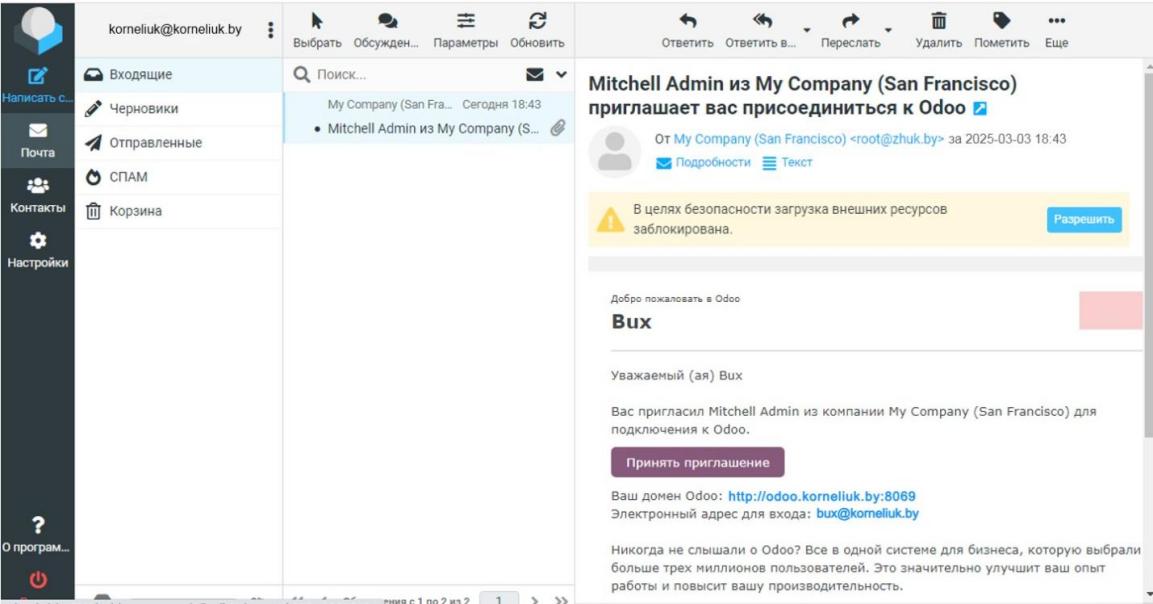
Права доступа: В разделе «Персонал» (Human Resources) выбираем «Сотрудник» (Officer) или подходящие права. В других приложениях, таких как «Администрирование», установите «Нет» или «Пользователь: Только собственные документы». Для этой лабораторной нужны права на HR (Персонал), чтобы видеть сотрудников.

Odoo должен автоматически отправляем письмо-приглашение на указанный адрес, используя настроенный на шаге 4 сервер исходящей почты.





1. **выполнить регистрацию пользователя в ERP-системе odoo, перейдя по ссылке в полученном на почтовый ящик письме и заполнив регистрационную форму.**

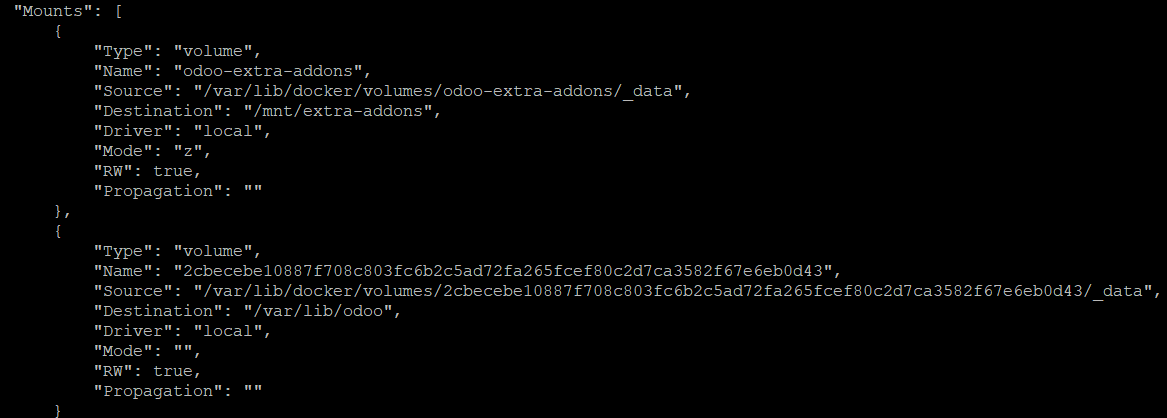


1. **определите пути в ОС Ubuntu и внутри контейнера odoo к папке, используемой odoo для размещения дополнительных модулей, с помощью команды**

docker inspect odoo | less

Найдите секцию «Mounts». Найдите запись, где Destination (Назначение) равно /mnt/extra-addons. Запомним путь Source (Источник). Это путь на вашей хостовой ВМ, где хранятся данные тома пользовательских модулей (/var/lib/docker/volumes/odoo-extra-addons/\_data). Это путь в ОС Ubuntu.

Путь внутри контейнера, используемый для пользовательских модулей, это тот, который мы указали в команде docker run: /mnt/extra-addons.



1. **запустите командную строку bash внутри контейнера командой**

docker exec -it odoo /bin/bash



1. **создайте заготовку модуля в папке для дополнительных модулей командой**

odoo scaffold test\_module /mnt/extra-addons/

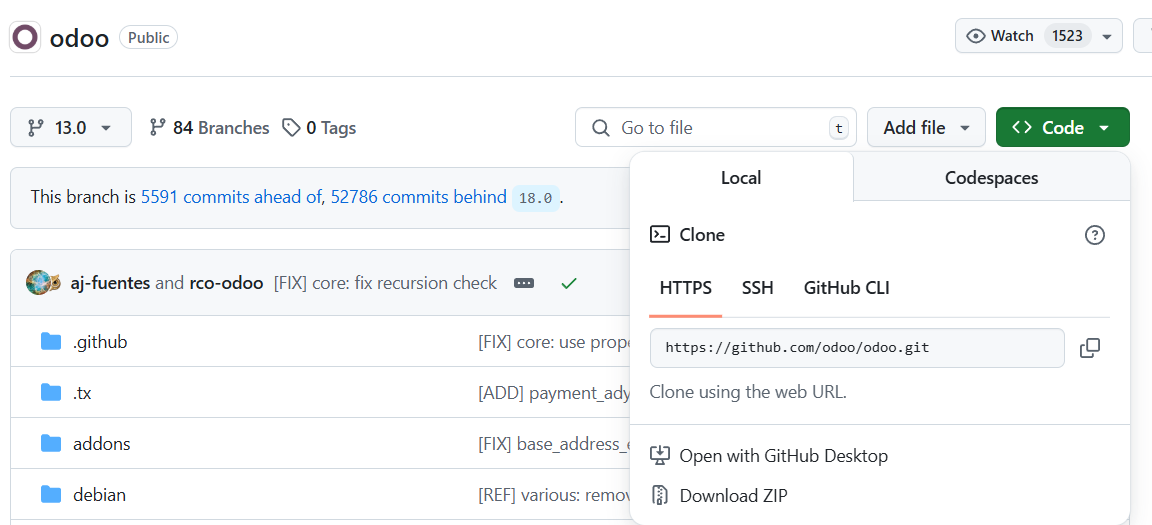
Эта команда использует встроенный инструмент Odoo для создания каркаса (scaffold), чтобы сгенерировать базовую структуру каталогов и шаблонные файлы для нового модуля с именем test\_module. Она пытается создать его в каталоге /mnt/extra-addons/ внутри контейнера.

1. **выйдите из командной строки внутри контейнера командой**

exit

1. **на физической машине для удобства редактирования исходного кода модуля скачайте исходный код odoo c github:**

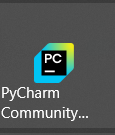
<https://github.com/odoo/odoo>



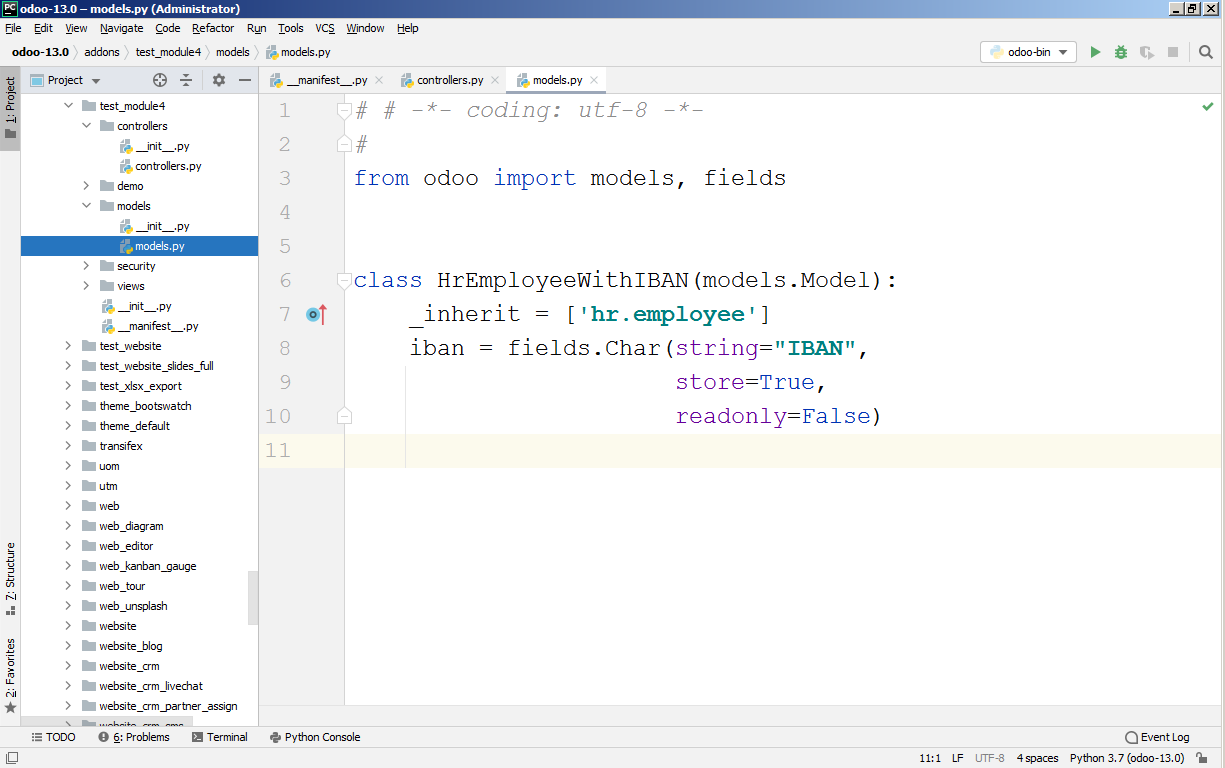
1. **распакоуйте скачанный исходный код в отдельную папку;**
2. **скопируйте по FTP с виртуальной машины созданную ранее заготовку модуля (папку)** test\_module **на физическую машину в папку** /addons **внутри папки с исходным кодом odoo;** **не перепутайте с папкой** /odoo/addons**;**

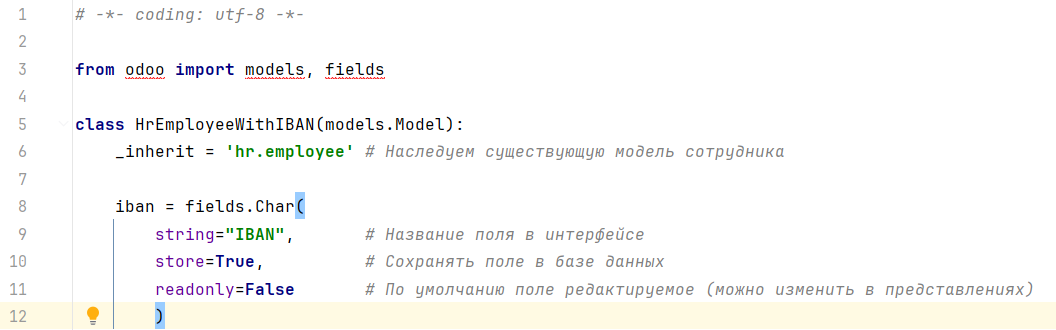
Перенести каталог test\_module (созданный odoo scaffold внутри контейнера, в /mnt/extra-addons/) с ВМ/контейнера на локальную машину разработки.

1. **для редактирования исходного кода модуля на языке Python целесообразно скачать и установить среду разработки PyCharm Community Edition, при установке поставьте галочку «Add “Open Folder as Project”»;**

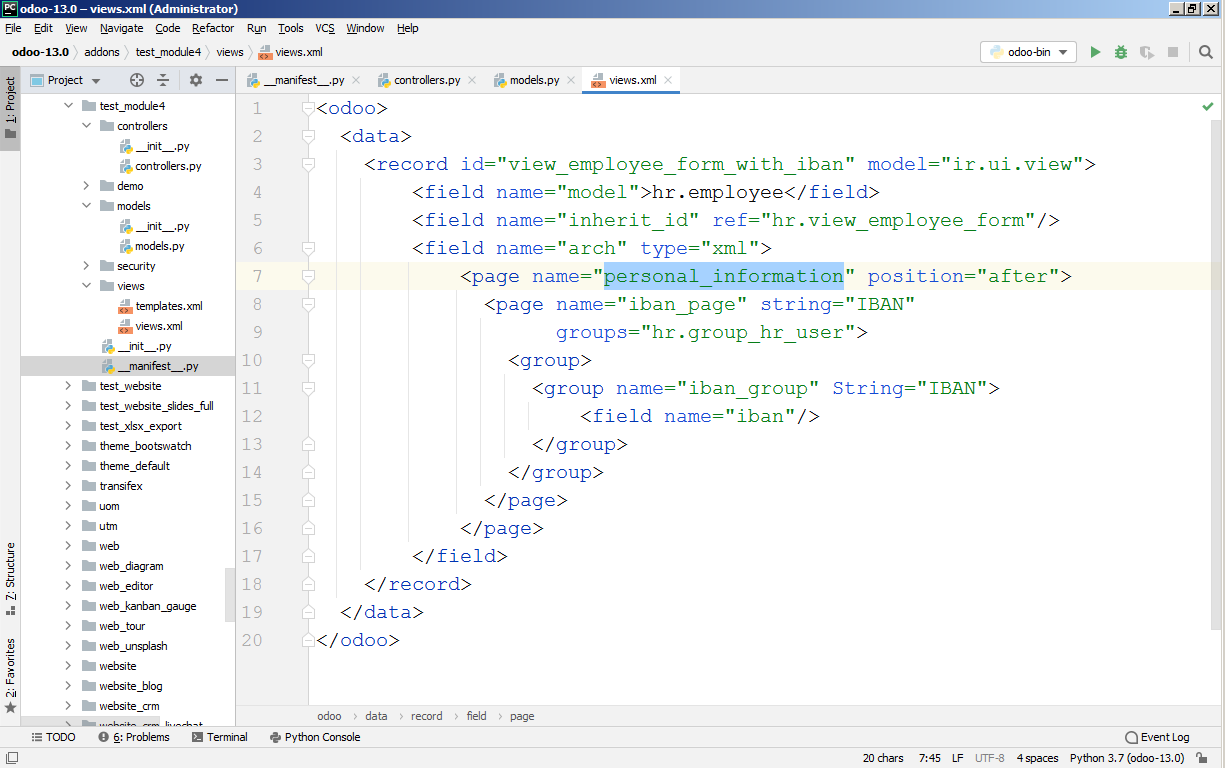


1. **нажмите правой кнопкой мыши по папке с исходным кодом odoo и выберите пункт «Open Folder as PyCharm Community Edition Project»;**
2. **на панели Project найдите заготовку модуля (папку);**
3. **в первую очередь необходимо обеспечить хранение номеров счетов работников с помощью модели, поэтому отредактируйте файл** /models/models.py **внутри папки с заготовкой модуля следующим образом для добавления поля** iban **к существующей модели** hr.employee:



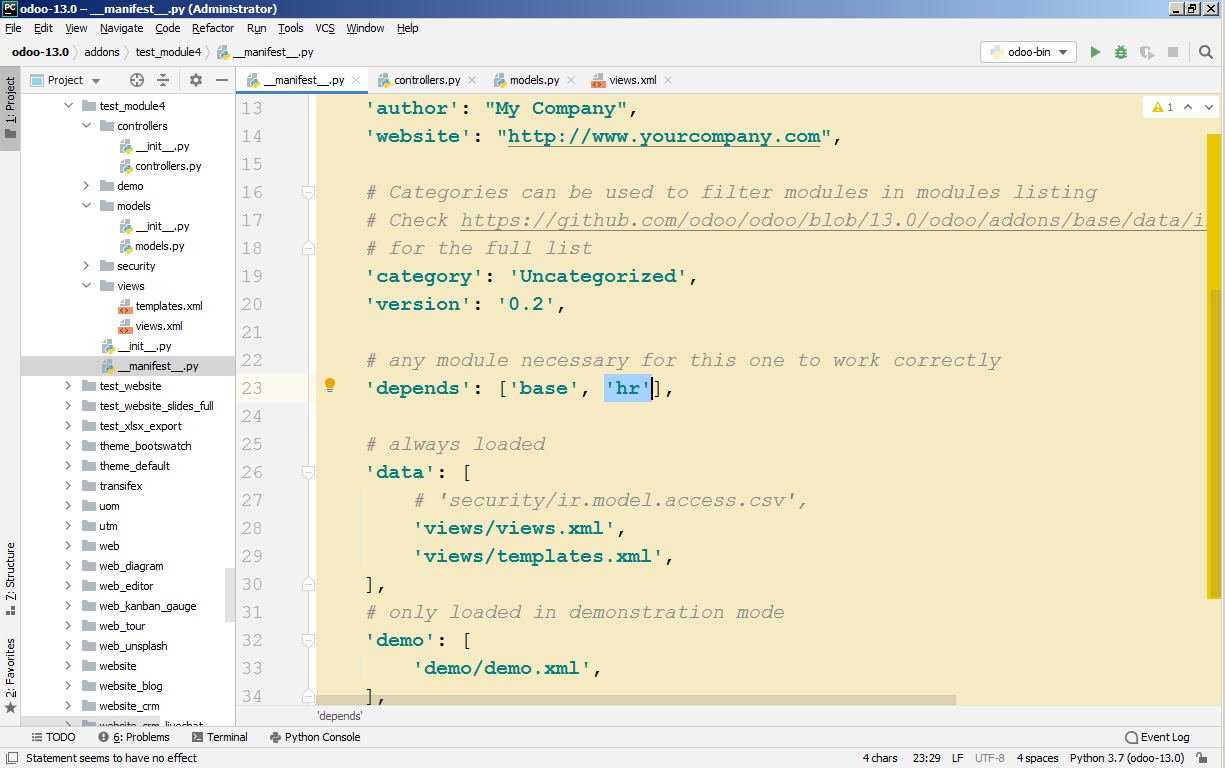


1. **для обеспечения возможности редактирования номера счета на веб-страницах odoo добавьте вкладку с полем для ввода/вывода номера счета в существующее представление** hr.view\_employee\_form, **например, после вкладки** personal\_information, **отредактировав файл** /views/views.xml **следующим образом:**



|  |
| --- |
| <odoo>  <data>  <record id=**"view\_employee\_form\_with\_iban"** model=**"ir.ui.view"**>  <field name=**"name"**>hr.employee.form.iban</field>  <field name=**"model"**>hr.employee</field>  <field name=**"inherit\_id"** ref=**"hr.view\_employee\_form"**/>  <field name=**"arch"** type=**"xml"**>  <!-- Находим вкладку **'personal\_information'** и добавляем нашу новую вкладку после нее -->  <xpath expr=**"//page[@name='personal\_information']"** position=**"after"**>  <page name=**"iban\_page"** string=**"Информация IBAN"**> <!-- Название вкладки -->  <group>  <group name=**"iban\_group"** string=**"Банковские реквизиты"**> <!-- Название группы -->  <field name=**"iban"**/> <!-- Поле для ввода IBAN -->  <!-- Ссылка будет добавлена на шаге 23 -->  <a href=**"/test\_module/belarusbank/export" class**=**"oe\_link"** target=**"\_blank"**>  <i **class**=**"fa fa-download"**/> Get salary list **for** Belarusbank  </a>  </group>  </group>  </page>  </xpath>  </field>  </record>  </data> </odoo> |

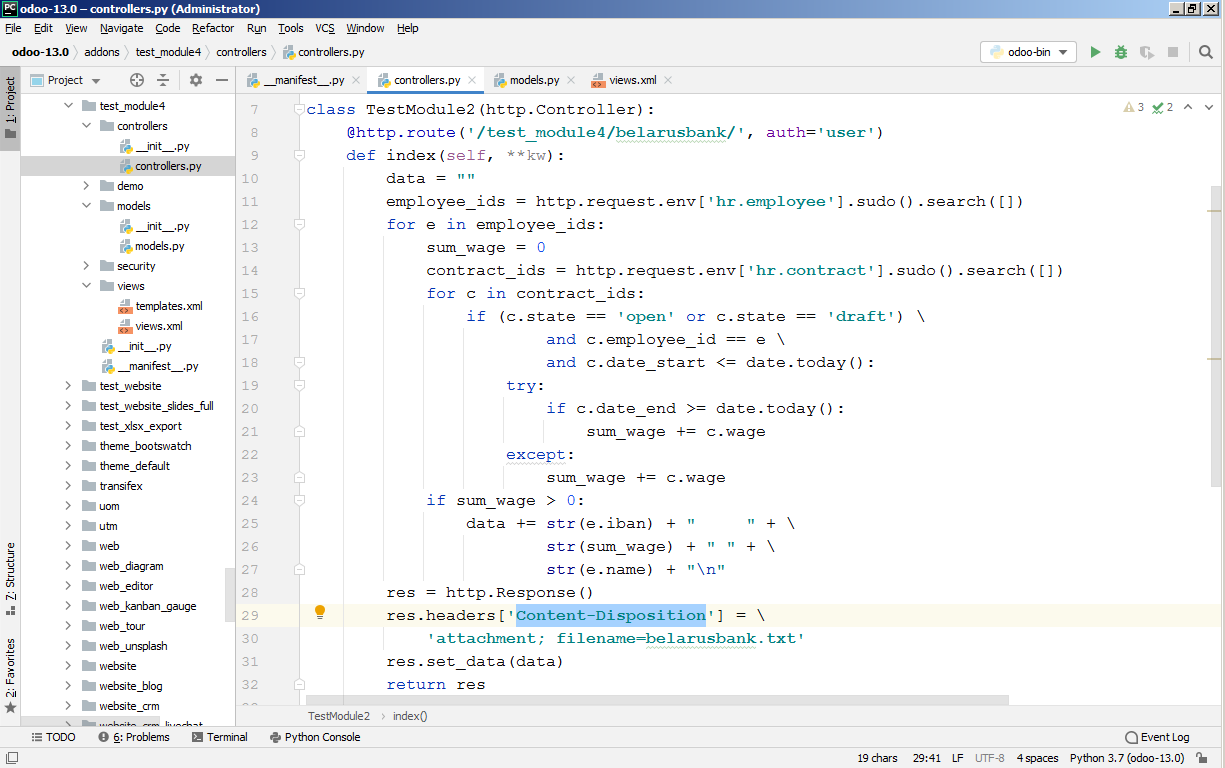
1. **поскольку наше расширяющее представление привязывается к модели из модуля hr, укажите данный модуль в списке зависимостей нашего модуля в файле** \_\_manifest\_\_.py:



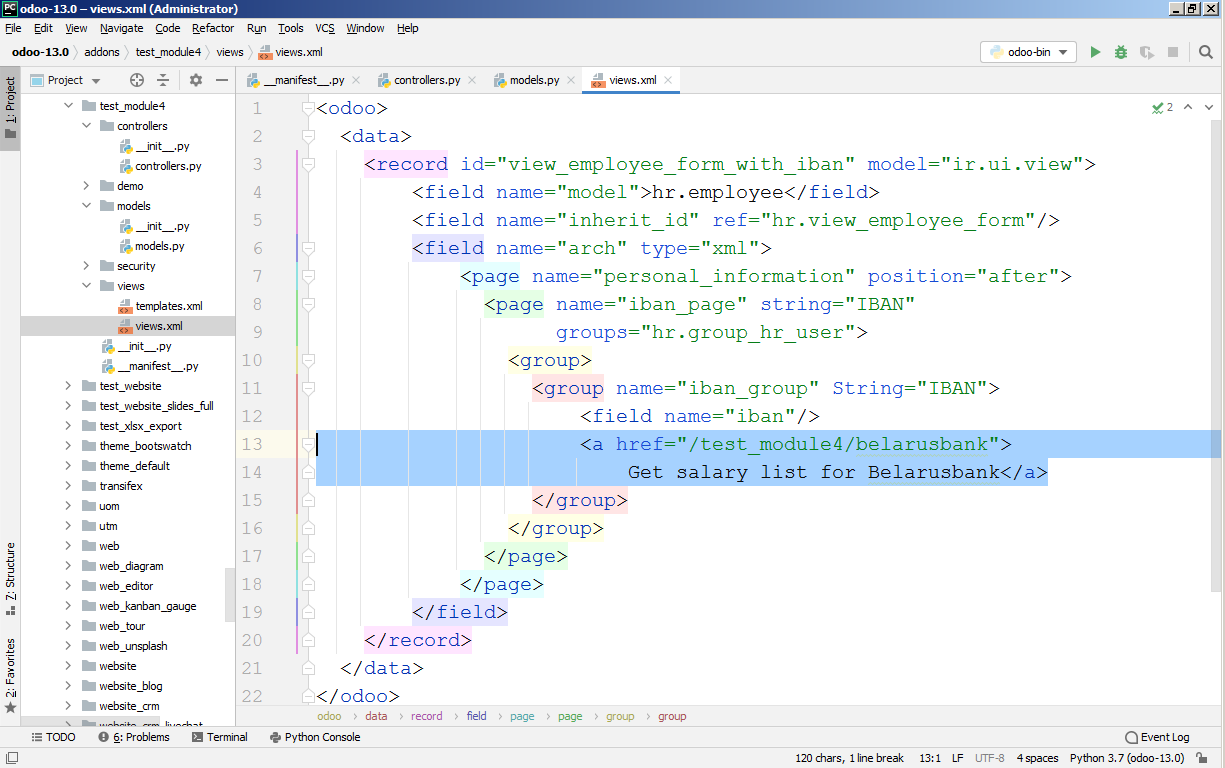
1. Открываем файл \_\_manifest\_\_.py непосредственно в папке test\_module.
2. Находим ключ 'depends'. Обычно он выглядит как 'depends': ['base'],.
3. Добавим 'hr' в список, так как наши изменения модели и представления зависят от модуля hr.
4. Убедимся, что номер версии обновлен:

|  |
| --- |
| *# -\*- coding: utf-8 -\*-* {  **'name'**: **"Тестовый Модуль IBAN"**, *# Дайте описательное имя* **'summary'**: **"""  Добавляет поле IBAN сотрудникам и функцию экспорта зарплат."""**,   **'description'**: **"""  - Добавляет поле IBAN в форму Сотрудника.  - Предоставляет контроллер для скачивания простого списка зарплат для Беларусбанка.  """**,   **'author'**: **"Лопатнюк Полина"**,**'website'**: **"http://www.yourcompany.com"**, *# Необязательно   # Категории используются для фильтрации модулей в списке  # См. https://github.com/odoo/odoo/blob/13.0/odoo/addons/base/data/ir\_module\_category\_data.xml  # для полного списка* **'category'**: **'Human Resources'**, *# Или Uncategorized* **'version'**: **'13.0.1.0.1'**, *# Версия Odoo.Major.Minor.Patch.Build   # Модули, необходимые для работы этого модуля* **'depends'**: [**'base'**, **'hr'**], *# Добавьте 'hr' сюда   # Всегда загружаемые данные* **'data'**: [  *# 'security/ir.model.access.csv', # Оставьте закомментированным, если не требуется* **'views/views.xml'**,  **'views/templates.xml'**, *# Оставьте, даже если пустой* ],  *# Данные, загружаемые только в демонстрационном режиме* **'demo'**: [  **'demo/demo.xml'**, *# Оставьте, даже если пустой* ],  **'installable'**: **True**,  **'application'**: **False**,  **'auto\_install'**: **False**, } |

1. **для импорта в клиент-банк требуется генерировать простой текстовый файл, поэтому требуется использовать не представление, а контроллер; создайте в файле /controllers/controllers.py контроллер с привязанным к определенному URL методом index, перебирающим всех работников, суммирующим для каждого работника оплату по всем действующим на текущую дату контрактам, формирующим файл из трех колонок, разделенных пробелами (IBAN, сумма выплат по контрактам, ФИО работника) и добавляющим перед файлом заголовок   
   Content-Disposition для загрузки этого файла, а не открытия в браузере:**



1. **добавьте в созданное ранее представление в файле /views/views.xml ссылку на метод созданного контроллера:**



1. **сохраните все отредактированные файлы;**
2. **передайте отредактированный модуль обратно на виртуальную машину, в папку для дополнительных модулей odoo; при отсутствии прав на запись выдайте права на папку командой chmod всем пользователям;**

Скопируем весь каталог test\_module с локальной машины разработки обратно на ВМ, поместив его в каталог, который Odoo отслеживает на предмет пользовательских аддонов

Узнаем UID/GID, под которым работает odoo в контейнере

docker exec odoo id -u

docker exec odoo id -g

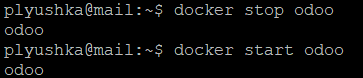
Меняем владельца на хостовом пути тома

sudo chown -R 101:101 /var/lib/docker/volumes/odoo-extra-addons/\_data/test\_module

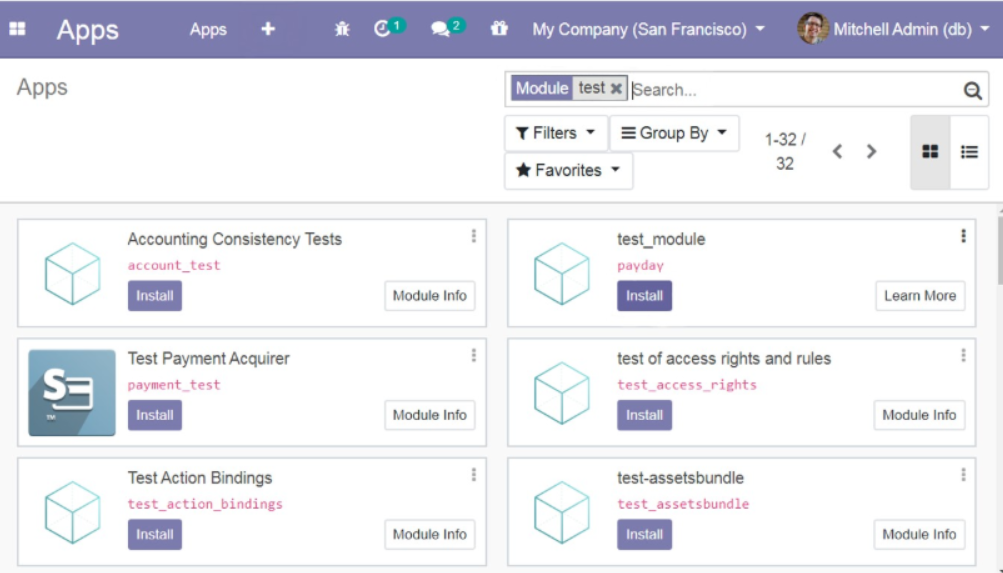
1. **для загрузки информации о новом модуле в odoo перезапустите контейнер командами**

docker stop odoo

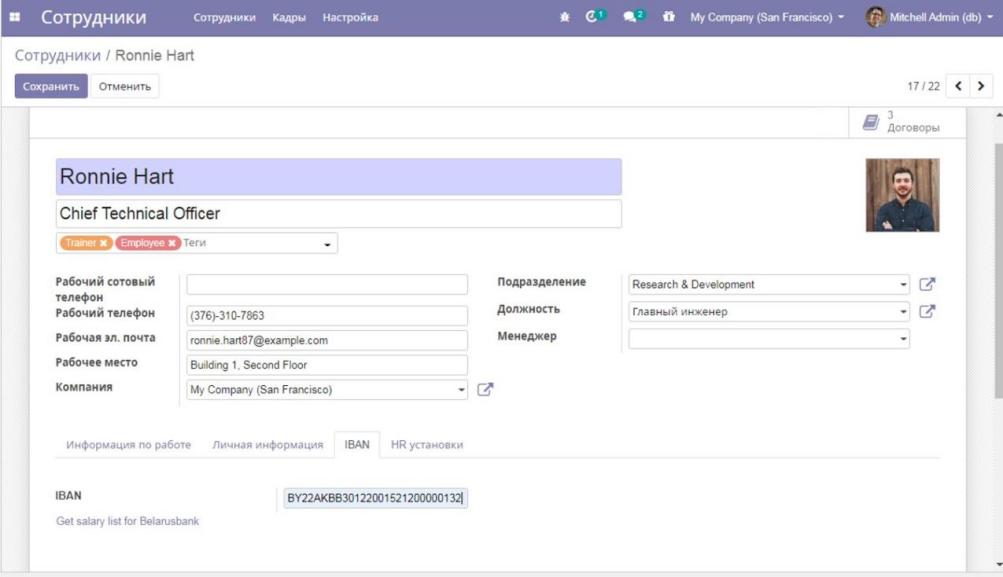
docker start odoo



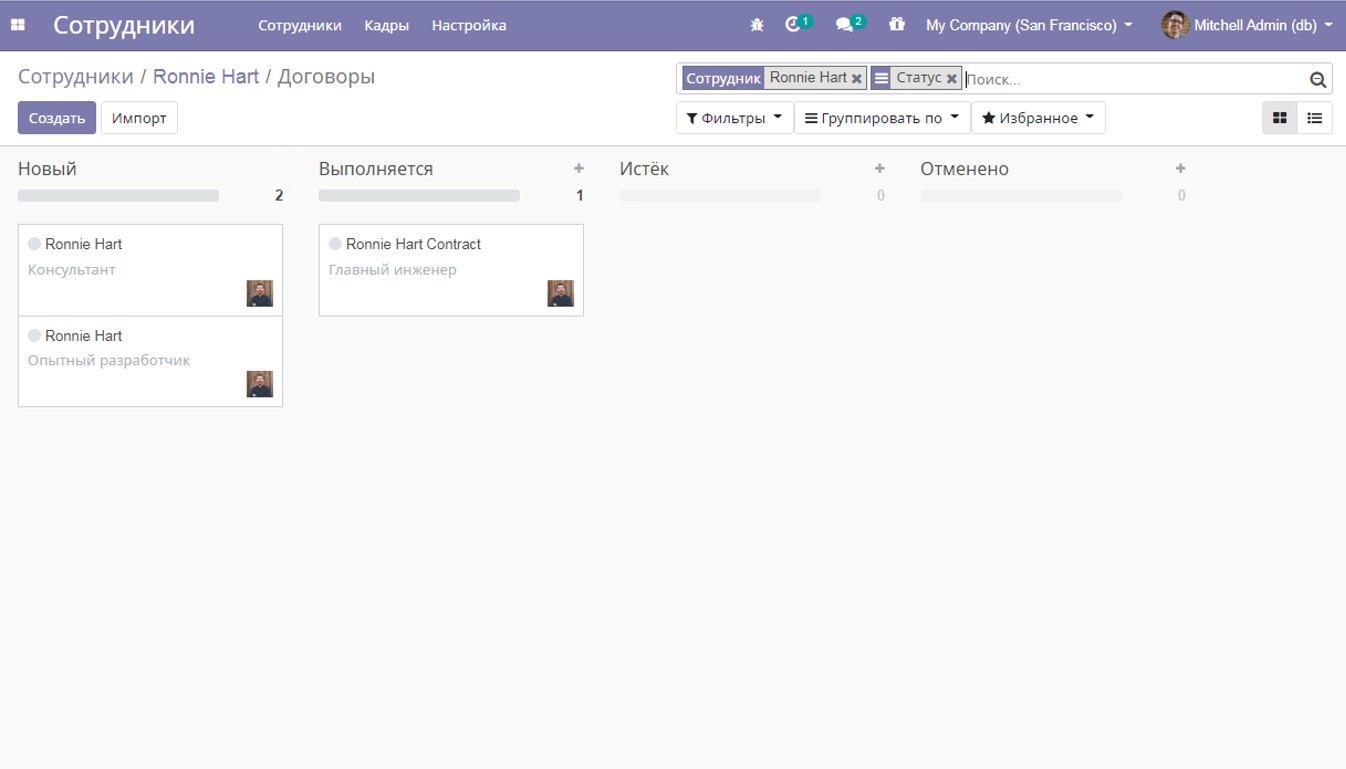
1. **для установки собственного модуля включите режим разработчика в настройках odoo;**
2. **установите созданный модуль через пункт главного меню «Приложения», найти свой модуль можно по названию с помощью панели поиска;**



1. **найдите сотрудников с действующими на текущую дату контрактами и внесите им IBAN;**



1. **найдите хотя бы одного сотрудника с несколькими действующими на текущую дату контрактами или добавьте новый контракт сотруднику с одним действующим контрактом;**



1. **нажмите на ссылку «Get salary list for Belarusbank» и убедитесь в корректности формирования списка на выплату зарплат: сотрудник с несколькими действующими контрактами не должен повторяться, для него должна быть указана сумма выплат по всем действующим контрактам.**

