

Лабораторная работа по теме № 10. Инфраструктура Docker.

Цель: освоить навыки развертывания контейнера из образа и его настройки в среде docker, передачи файлов между операционной системой и контейнером docker, запуска интерактивных приложений внутри контейнера и управления контейнером.

Задачи:

1. на виртуальной машине через утилиту apt установить клиентский пакет docker и серверный пакет dockerd;
2. . при использовании университетской сети сконфигурировать dockerd на использование прокси-сервера 172.16.0.101:3128;
3. выполнить установку ERP-системы odoo с помощью docker и инструкций на сайте hub.docker.com, согласиться на внесение в базу данных демонстрационных данных;
4. сконфигурировать ERP-систему odoo на использование SMTP-сервера postfix, входящего в состав пакета iRedMail, для отправки e-mail;
5. сконфигурировать SMTP-сервер postfix на прием писем из docker-контейнера с ERP-системой odoo;
6. создать почтовый ящик на почтовом сервере с помощью панели iRedAdmin;
7. создать пользователя в ERP-системе odoo, назначить ему права на использование только бухгалтерских компонентов, выслать подтверждение на созданный почтовый ящик;
8. выполнить регистрацию пользователя в ERP-системе odoo, перейдя по ссылке в полученном на почтовый ящик письме и заполнив регистрационную форму.
9. определите пути в ОС Ubuntu и внутри контейнера odoo к папке, используемой odoo для размещения дополнительных модулей, с помощью команды

docker inspect odoo | less

10. запустите командную строку bash внутри контейнера командой

docker exec -it odoo /bin/bash

11. создайте заготовку модуля в папке для дополнительных модулей командой
odoo scaffold test_module /mnt/extra-addons/

12. выйдите из командной строки внутри контейнера командой
exit

13. на физической машине для удобства редактирования исходного кода модуля скачайте исходный код odoo с github:

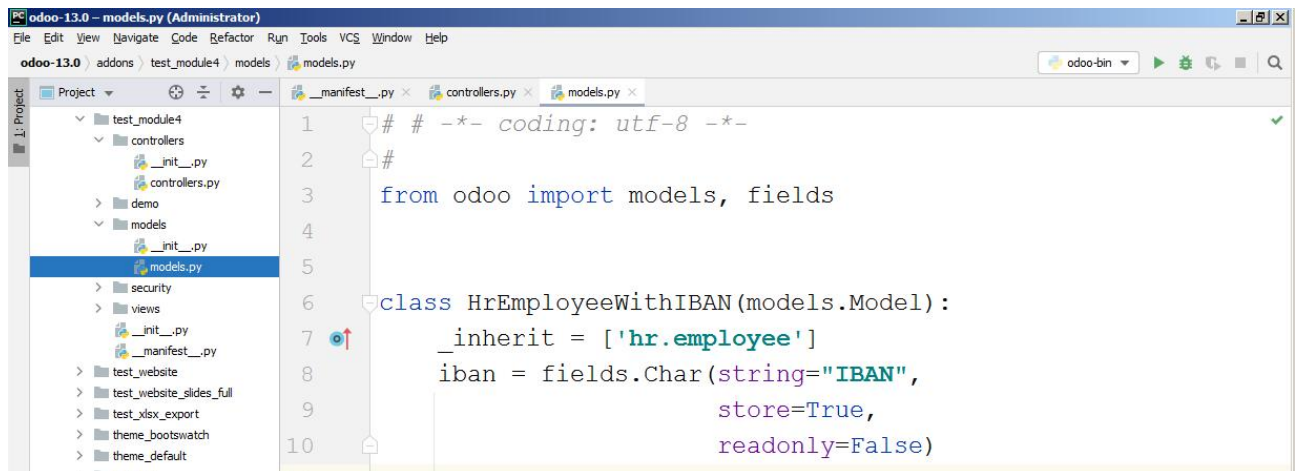
<https://github.com/odoo/odoo>

14. распакуйте скачанный исходный код в отдельную папку;
15. 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;**

16. 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”»;

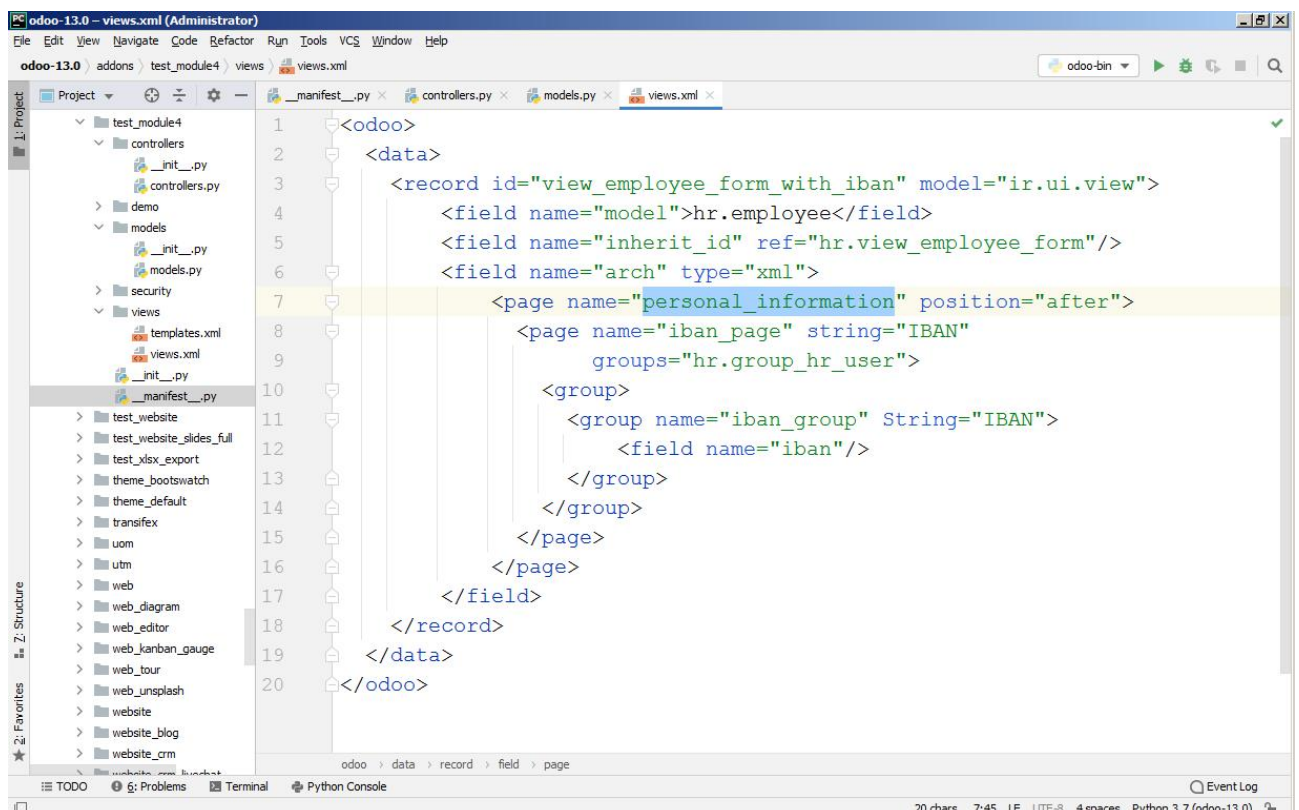
17. нажмите правой кнопкой мыши по папке с исходным кодом odoo и выберите пункт «Open Folder as PyCharm Community Edition Project»;

18. на панели Project найдите заготовку модуля (папку);
19. 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**:



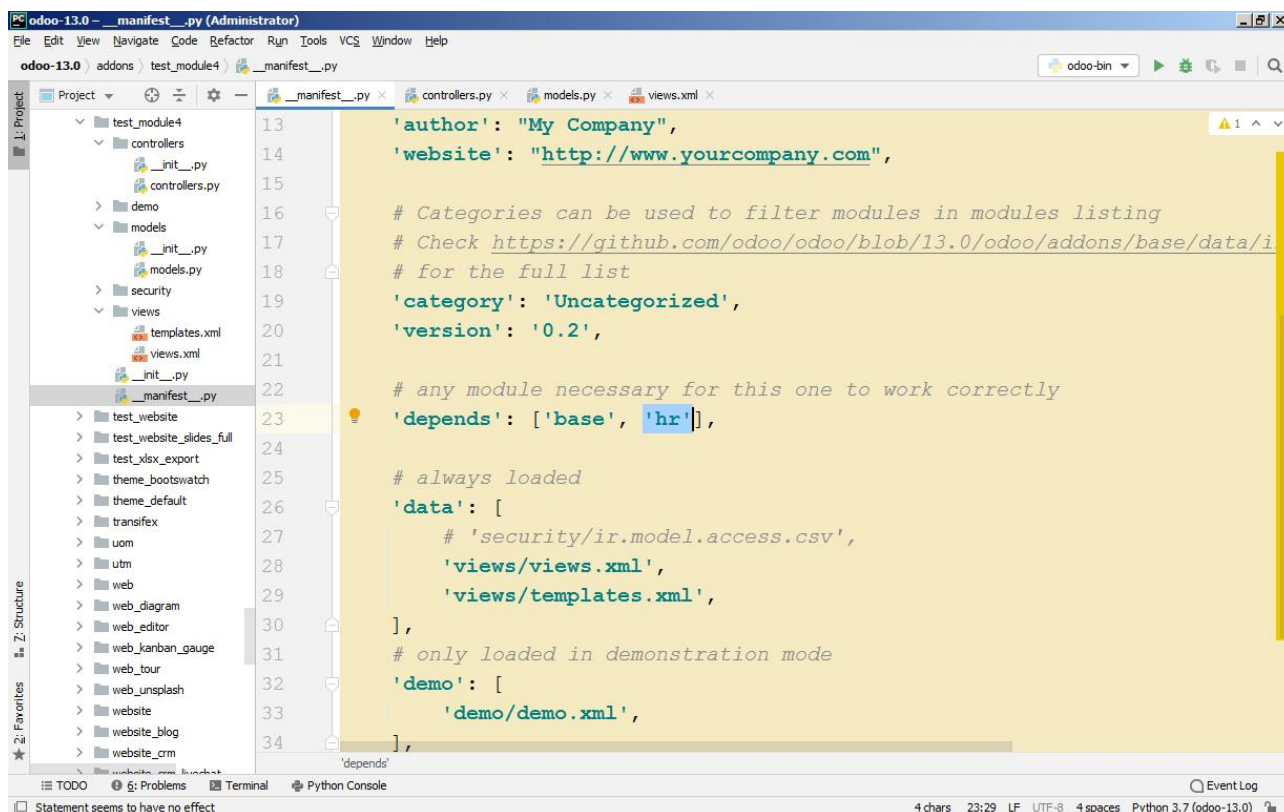
```
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields
4
5
6  class HrEmployeeWithIBAN(models.Model):
7      _inherit = ['hr.employee']
8      iban = fields.Char(string="IBAN",
9                          store=True,
10                         readonly=False)
```

20. 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:



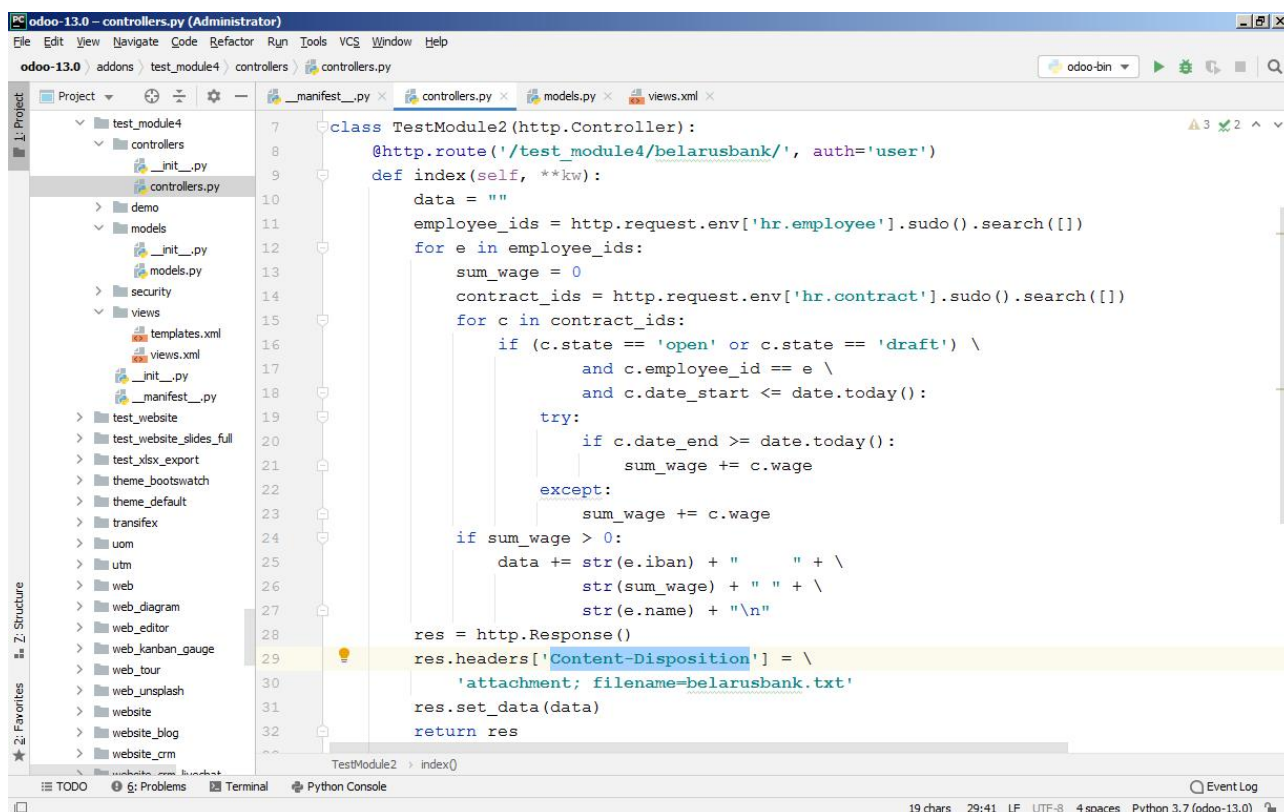
```
1  <odoo>
2      <data>
3          <record id="view_employee_form_with_iban" model="ir.ui.view">
4              <field name="model">hr.employee</field>
5              <field name="inherit_id" ref="hr.view_employee_form"/>
6              <field name="arch" type="xml">
7                  <page name="personal_information" position="after">
8                      <page name="iban_page" string="IBAN"
9                          groups="hr.group_hr_user">
10                         <group>
11                             <group name="iban_group" String="IBAN">
12                                 <field name="iban"/>
13                             </group>
14                         </group>
15                     </page>
16                 </page>
17             </field>
18         </record>
19     </data>
20 </odoo>
```

21. поскольку наше расширяющее представление привязывается к модели из модуля **hr**, укажите данный модуль в списке зависимостей нашего модуля в файле **__manifest__.py**:



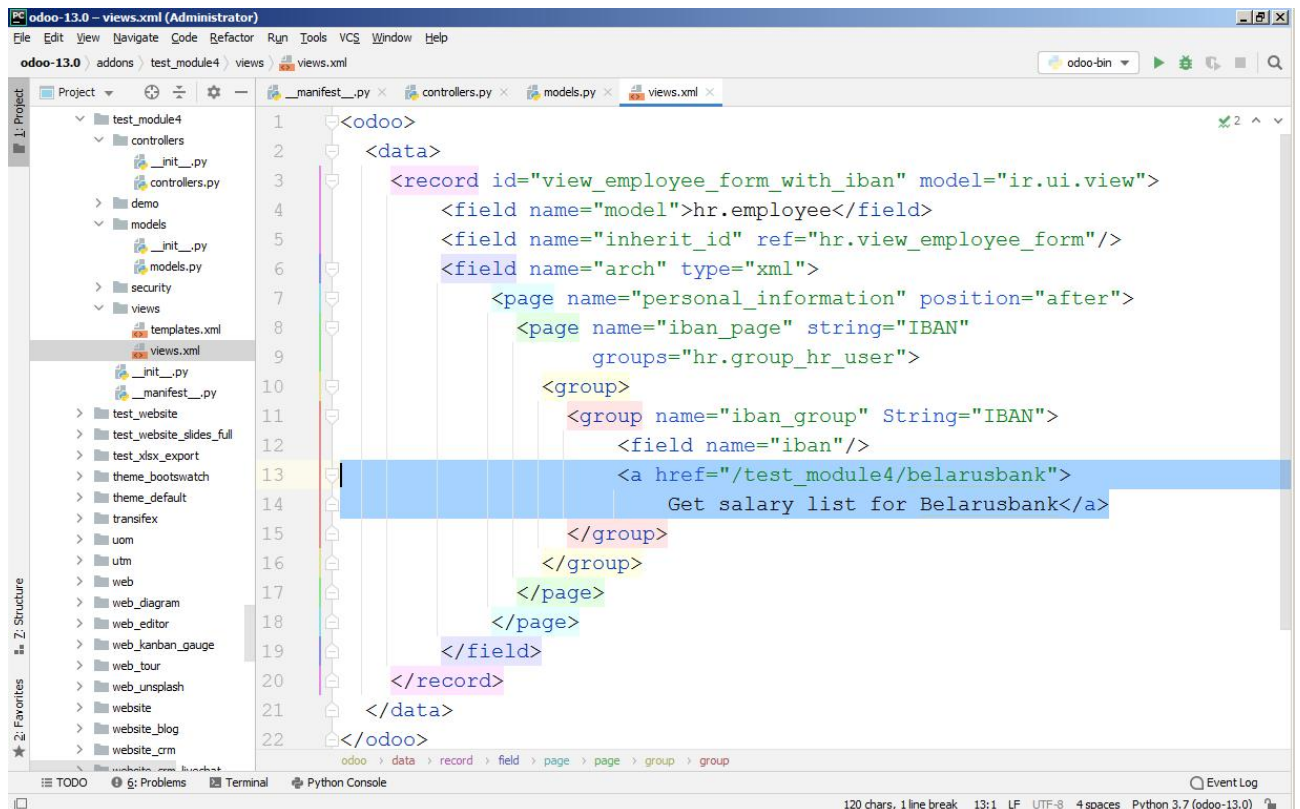
```
13 'author': "My Company",
14 'website': "http://www.yourcompany.com",
15
16 # Categories can be used to filter modules in modules listing
17 # Check https://github.com/odoo/odoo/blob/13.0/odoo/addons/base/data/i
18 # for the full list
19 'category': 'Uncategorized',
20 'version': '0.2',
21
22 # any module necessary for this one to work correctly
23 'depends': ['base', 'hr'],
24
25 # always loaded
26 'data': [
27     # 'security/ir.model.access.csv',
28     'views/views.xml',
29     'views/templates.xml',
30 ],
31 # only loaded in demonstration mode
32 'demo': [
33     'demo/demo.xml',
34 ],
```

22. 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:



```
7 class TestModule2(http.Controller):
8     @http.route('/test_module4/belarusbank/', auth='user')
9     def index(self, **kw):
10         data = ""
11         employee_ids = http.request.env['hr.employee'].sudo().search([])
12         for e in employee_ids:
13             sum_wage = 0
14             contract_ids = http.request.env['hr.contract'].sudo().search([])
15             for c in contract_ids:
16                 if (c.state == 'open' or c.state == 'draft') \
17                     and c.employee_id == e \
18                     and c.date_start <= date.today():
19                     try:
20                         if c.date_end >= date.today():
21                             sum_wage += c.wage
22                     except:
23                         sum_wage += c.wage
24             if sum_wage > 0:
25                 data += str(e.iban) + " " + \
26                     str(sum_wage) + " " + \
27                     str(e.name) + "\n"
28         res = http.Response()
29         res.headers['Content-Disposition'] = \
30             'attachment; filename=belarusbank.txt'
31         res.set_data(data)
32         return res
```

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



24. сохраните все отредактированные файлы;

25. 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;

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

docker stop odoo

docker start odoo

27. для установки собственного модуля включите режим разработчика в настройках odoo;

28. установите созданный модуль через пункт главного меню «Приложения», найти свой модуль можно по названию с помощью панели поиска;

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

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

31. 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.

Порядок защиты лабораторной работы.

1. Продемонстрировать в браузере вход в ERP-систему odoo под созданным пользователем.
2. Продемонстрировать в браузере страницу сотрудника с введенным IBAN.

3. Продемонстрировать в браузере скачивание текстового файла при нажатии на ссылку «Get salary list for Belarusbank».
4. Продемонстрировать содержимое скачанного текстового файла, проверить соответствие номера счета для сотрудника и посчитанной суммы выплат по контрактам.