

Отчеты и печатные формы

Введение.....	1
Wkhtmltopdf.....	1
Создание отчета.....	2
Как сделать?.....	5
Разметку	5
Вставить поле	5
Вставить картинку	5
Присвоить значение переменной.....	5
Запустить цикл по строкам	5
Сколько страниц в отчете	6
Отобразить что-то в зависимости от условий	6
Задание	6

Введение

Построение отчетов и печатных форм документов в odoo построено на технологии QWEB. Это собственная разработка odoo, нигде, кроме odoo, она не встречается. Основная идея заключается в том, чтобы создать шаблон, в котором будет совмещена HTML разметка документа и подсистема обращения к пользовательским данным для наполнения этих HTML шаблонов.

Используется в odoo для создания отчетов и для создания шаблонов веб-сайта.

Разработка печатной формы, далее ПФ, состоит из следующих этапов:

1. Подготовка максимально чистой HTML разметки будущего документа. Вы можете воспользоваться любым редактором.
2. Внедрение в HTML элементов QWEB.
3. Создание XML, в котором будет наш шаблон и другие параметры печатной формы, такие как размер бумаги, качество печати, связанная модель и др.

Как опытный программист, работающий с open source, первым делом Вы пойдете скорее всего в модуль sale, там найдете отчеты и попытаете разобраться, что к чему. И потерпите фиаско. Слишком уж много нюансов.

Давайте разберем простейший отчет с нуля. Но сначала несколько технических замечаний

Wkhtmltopdf

Каждая ПФ — это веб-страница. Для перевода в формат PDF, удобный для пересылки и печати на устройствах, используется утилита wkhtmltopdf.

Используйте строго ту версию, которая совместима с вашей версией odoo. Например, для 18 ядра нужна версия 0.12.6.

Проверьте вашу версию:

```
$ wkhtmltopdf --version
```

Если она отличается от 0.12.6, удалите текущую и установите нужную (вместе с зависимостями).

Создание отчета

Файлы отчетов должны быть размещены в папке reports вашего модуля.

Давайте представим, что у нас есть модуль «Коммерческие предложения». Это просто хранилище строк с полями в HTML формате, которые мы хотим красиво вывести в PDF.

Давайте в папке reports создадим файл offer.xml:

```
<?xml version="1.0"?>
<odoo>
  <record id="action_mtaoffer_report" model="ir.actions.report">
    <field name="name">Коммерческое предложение</field>
    <field name="model">sale.order</field>
    <field name="report_type">qweb-xml</field>
    <field name="report_name">mta_base.report_offer_template </field>
    <field name="report_file"> mta_base.report_offer_template </field>
    <field name="binding_model_id" ref="sale.model_sale_order"/>
    <field name="print_report_name">'%s' % (object.name)</field>
  </record>
</odoo>
```

record id — идентификатор вашего отчета в системе. По нему можно будет обратиться из кода Python.

name — название

model — модель, к которой привязывается отчет, и откуда он будет брать данные.

report_type — бывает qweb-pdf и qweb-html. Отличается тем, что конвертация в PDF в первом случае происходит автоматически, а во втором по кнопке «Печать», а до этого отчет показывается в HTML формате. Удобно для отладки.

report_name="mta_base.report_offer_template" имя отчета в формате *модуль.имя_отчета*. Это имя указывает на то, какой шаблон отчета используется.

binding_model_id="model_sale_order" — наименование модели, из которой будет воспроизводиться печать. Также при отображении записи этой модели в меню «Печать» появится этот отчет.

print_report_name — это не обязательное поле, которое используется для именования сгенерированного файла отчета с использованием выражений Python в соответствии с необходимостью.

Мы рассмотрели так называемую заголовочную запись. Она говорит системе, что у нас есть вот такой отчет. А сам шаблон где-то ниже или в других файлах.

Дальше есть смысл в файле отчета сделать свой формат бумаги:

```
<record id="paperformat_a4offer" model="report.paperformat">
  <field name="name">A4</field>
  <field name="default" eval="True" />
  <field name="format">A4</field>
  <field name="page_height">0</field>
  <field name="page_width">0</field>
  <field name="orientation">Portrait</field>
  <field name="margin_top">70</field>
  <field name="margin_bottom">30</field>
  <field name="margin_left">7</field>
  <field name="margin_right">7</field>
  <field name="header_line" eval="False" />
  <field name="header_spacing">60</field>
  <field name="dpi">90</field>
</record>
```

Здесь комментарии излишни. Вы можете поиграть этими параметрами для достижения наилучшего внешнего вида вашей печатной формы.

Если Вы не указали свой формат бумаги, можете указать любой существующий или вообще его не указывать. Система возьмет формат бумаги по умолчанию, который Вы задали в Общих настройках.

Теперь нужно указать системе, что наш отчет имеет такой тип бумаги:

```
<record id="action_mtaoffer_report" model="ir.actions.report">
  <field name="paperformat_id" ref="paperformat_a4offer"/>
</record>
```

action_mtaoffer_report — ID отчета

paperformat_a4offer — ID формата бумаги

Повторюсь, что манипуляции со своим форматом бумаги не обязательны, но Вам нужно знать, как это делается.

И теперь переходим к написанию шаблона. Рассмотрим последовательно. Код отчета будет отдельным файлом к этому уроку.

```
<template id="mta_base.report_offer_template">
mta_base.report_offer_template — то самое имя отчета
```

```
<t t-call="web.html_container">
```

Вызываем базовый контейнер. Это обязательная часть, иначе нельзя будет сформировать веб-страницу.

```
<t t-foreach="docs" t-as="o">
```

А вот тут подробнее. Пошел жесткий QWEB.
Структуры QWEB, внедренные в ваш html часто имеют формат:

```
<t команда=имя команды> </t>
```

Команды бывают:

t-foreach - запускает цикл

t-set установить значение переменной

t-esc — значение с очисткой от возможного html

t-field — значение соответствующего поля модели

t-call вызов другого шаблона

t-as присваивание имени

t-value присваивание значения переменной

t-options — опции

t-if проверка условия

Вы обратили внимание, что вроде бы имеются внутренние переменные, например, docs. Да, они есть, и к ним можно обращаться:

docs — содержат набор документов, а равно записей модели, для которых вызван отчет. Отчет ведь можно вызывать не для одной записи, а для многих.

doc_ids идентификаторы записей

doc_model — модель данных

time — это объект модуля time из Python

user — объект типа res.users, содержащий пользователя, под которым запускается ПФ

res_company — объект res.company, из под которой работает пользователь

Идем дальше:

```
<t t-call="web.external_layout"> или  
  <t t-call="web.basic_layout">
```

Вызываем шаблон отчета, но не наш, а шаблон внешнего вида.

Если хотим красивый со встроенным header и footer, берем **web.external_layout**.

Если будем делать header и footer сами, то пишем:

```
<t t-call="web.basic_layout">
```

Далее мы должны задать контейнер:

```
<div class="page">
```

Внутри этого контейнера и будет весь наш контент.

Мы можем сразу после открытия этого контейнера вставить свои стили:

```
<style type="text/css">
```

```
  @page { margin-right: 1.5cm; margin-top: 1.25cm; margin-bottom: 1.25cm }  
  p { margin-bottom: 0.21cm; direction: ltr; color: #000000; orphans: 2; widows: 2 }  
  p.western { font-family: "Times New Roman", serif; font-size: 12pt; so-language: ru-RU }  
  p.cjk { font-family: "Times New Roman", serif; font-size: 12pt; so-language: zh-CN }  
  p.ctl { font-family: "Times New Roman", serif; font-size: 12pt; so-language: ar-SA }  
  a:link { color: #0000ff }
```

```
</style>
```

Дальше, мы можем задать header:

```
<div class="header">
```

Внутри напишем то, что будет в верхнем колонтитуле.

После этого напишем наш отчет, и затем зададим секцию с нижним колонтитулом:

```
<div class="footer">
```

Есть модули, которые расширяют управление колонтитулами, например, не показывать на первой, не показывать на последней и пр.

Со структурой файла шаблона Вы познакомитесь в примере, а сейчас перейдем к частым вопросам

Как сделать?

Разметку

Используйте **bootstrap**. И любые html теги из html5.

```
<div class="row">  
<div class="col-xs-3 pull-right">
```

Почитайте подробно про табличные системы Bootstrap.

Вставить поле

```
<span t-field="line.name_html"/>
```

Вставить картинку

Статическая:

```

```

Динамическая (из поля):

```
<span t-field="o.mt_contractid.company_id.chief_id.facsimile" t-field-options='{ "widget":  
"image", "style": "width: 90px; height: 90px" }' />
```

Присвоить значение переменной

```
<t t-set="num" t-value="num+1"/>
```

Запустить цикл по строкам

```
<t t-foreach="o.line_ids" t-as="line">
```

Сколько страниц в отчете

`` - отображает текущий номер страницы

`` - отображает сколько всего страниц в документе

Эти штуки полезны для footer, когда надо написать Страница N из M.

Отобразить что-то в зависимости от условий

```
<t t-if="not o.mt_contractid.stamp">
    если нет печати, то показываем что-то
</t>
```

Задание

Сделайте печатную форму, которая выводит для собрания из модуля snt-rassvet, нумерованный список участников.

Например так:

Собрание №1 от 20.01.2020.

Состав участников:

1. Иванов Иван.

2. Петров Петр.