onnet

2021
VIE

# Odoo Qweb PDF Report

www.on.net.vn
www.arrowhitech.com

onnet
BY AHT TECH JSC

# Table of contents




onnet
BY AHT TECH JSC

# Reporting in Odoo

Odoo uses a report engine based on **QWeb Templates**, **Twitter Bootstrap** and **Wkhtmltopdf**.

Reports are written in HTML/QWeb, like website views in Odoo.
Reports are declared using a **report action**, and a **Report template** for the action to use.

A report is a combination two elements:

- an **ir.actions.report** which configures various basic parameters for the report

```
<record id="account_invoices" model="ir.actions.report">
   <field name="name">Invoices</field>
   <field name="model">account.invoice</field>
   <field name="report_type">qweb-pdf</field>
   <field name="report_name">account.report_invoice</field>
   <field name="report_file">account.report_invoice</field>
   <field name="attachment_use" eval="True"/>
   <field name="attachment">(object.state in ('open','paid')) and
      ('INV'+(object.number or '').replace('/','')+'.pdf')</field>
   <field name="binding_model_id" ref="model_account_invoice"/>
   <field name="binding_type">report</field>
</record>
```

# Reporting in Odoo

Odoo uses a report engine based on **QWeb Templates**, **Twitter Bootstrap** and **Wkhtmltopdf**.

Reports are written in HTML/QWeb, like website views in Odoo.
Reports are declared using a **report action**, and a **Report template** for the action to use.

A report is a combination two elements:

- an **ir.actions.report** which configures various basic parameters for the report
- a standard **qweb view** for the actual report

```
<t t-call="web.html_container">
    <t t-foreach="docs" t-as="o">
        <t t-call="web.external_layout">
            <div class="page">
                <h2>Report title</h2>
            </div>
        </t>
    </t>
</t>
```

# Reporting in Odoo

Odoo uses a report engine based on **QWeb Templates**, **Twitter Bootstrap** and **Wkhtmltopdf**.

Reports are written in HTML/QWeb, like website views in Odoo.
Reports are declared using a **report action**, and a **Report template** for the action to use.

A report is a combination two elements:
- an **ir.actions.report** which configures various basic parameters for the report
- a standard **qweb view** for the actual report

The PDF rendering itself is performed by **wkhtmltopdf**.
It is possible to specify a **Paper Format** for the report report.

# Report Action (ir.actions.report)

## Menus & Actions

The **actions** define the behavior of the system in response to the actions of the users ; login of a new user, double-click on an invoice, click on the action button, ...

There are different types of simple actions:

- Window
- Report
- Wizard
- ...

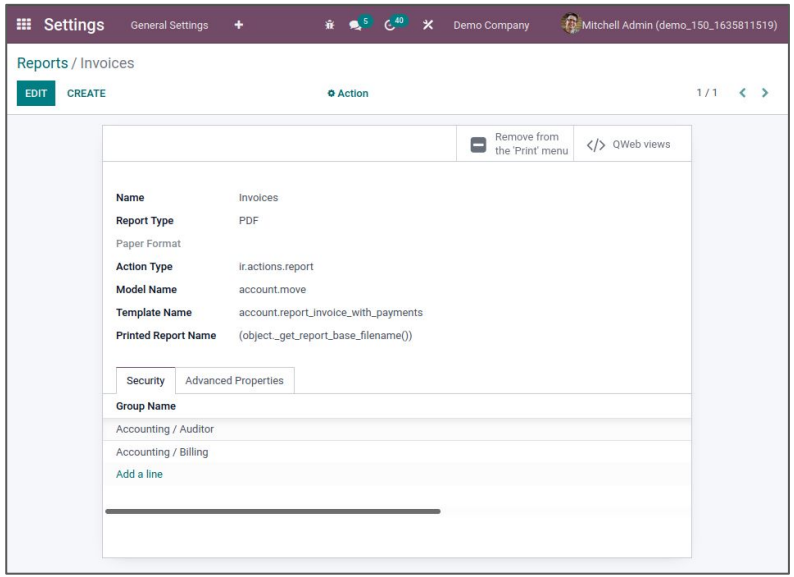
Actions can be triggered in three ways:

- by clicking on menu items (linked to specific actions)
- by clicking on buttons in views (if these are connected to actions)
- as contextual actions on object

# Report Action (ir.actions.report)

Triggers the printing of a report

Actions can be stored in the database or returned directly as dictionaries




```
<record id="account_invoices" model="ir.actions.report">
    <field name="name">Invoices</field>
    <field name="model">account.invoice</field>
    <field name="report_type">qweb-pdf</field>
    <field name="report_name">account.report_invoice</field>
    <field name="report_file">account.report_invoice</field>
    <field name="attachment_use" eval="True"/>
    <field name="attachment">(object.state in ('open','paid')) and
        ('INV'+(object.number or '').replace('/','')+'.pdf')</field>
    <field name="binding_model_id" ref="model_account_invoice"/>
    <field name="binding_type">report</field>
</record>
```

# Report Action (ir.actions.report)

Fields:

- name
- model: the model your report will be about
- report_type: HTML, PDF, Text
- report_name: the name (**external id**) of the qweb template used to render the report
- print_report_name: python expression defining the name of the report
- groups-id: **Many2many** field to the groups allowed to view/use the current report
- attachment-use: report will be re-printed from the stored report instead of being re-generated every time.
- attachment: python expression that defines the name of the report; the record is accessible as the variable **object**

```
<record id="account_invoices" model="ir.actions.report">
    <field name="name">Invoices</field>
    <field name="model">account.invoice</field>
    <field name="report_type">qweb-pdf</field>
    <field name="report_name">account.report_invoice</field>
    <field name="report_file">account.report_invoice</field>
    <field name="attachment_use" eval="True"/>
    <field name="attachment">(object.state in ('open','paid')) and
        ('INV'+(object.number or '').replace('/','')+'.pdf')</field>
    <field name="binding_model_id" ref="model_account_invoice"/>
    <field name="binding_type">report</field>
</record>
```

# Report Template

A minimal template would look like:

- Calling **web.external_layout** will add the default header and footer on your report
- The PDF body will be the content inside the `<div class="page">`

By default, the rendering context will also expose the following items:

- docs: records for the current report
- doc_ids: list of the ids for the docs records
- doc_model: model for the docs records

```
<template id="report_invoice">
    <t t-call="web.html_container">
        <t t-foreach="docs" t-as="o">
            <t t-call="web.external_layout">
                <div class="page">
                    <h2>Report title</h2>
                    <p>This object's name is
                        <span t-field="o.name"/>
                    </p>
                </div>
            </t>
        </t>
    </t>
</template>
```

# Qweb Template

**QWeb** is the primary **templating** engine used by Odoo

It is an XML templating engine and used mostly to generate HTML fragments and pages.

Template directives are specified as XML attributes prefixed with **t-**

- t-if, t-elif, t-else

```
<div>
    <p t-if="user.birthday == today()">Happy birthday!</p>
    <p t-elif="user.login == 'root'">Welcome master!</p>
    <p t-else="">Welcome!</p>
</div>
```

# Qweb Template

**QWeb** is the primary **templating** engine used by Odoo

It is an XML templating engine and used mostly to generate HTML fragments and pages.

Template directives are specified as XML attributes prefixed with **t-**

- t-if, t-elif, t-else
- t-out

```
<p><t t-out="value"/></p>
```

Will be rendered as:

```
<p>42</p>
```

# Qweb Template

**QWeb** is the primary **templating** engine used by Odoo

It is an XML templating engine and used mostly to generate HTML fragments and pages.

Template directives are specified as XML attributes prefixed with **t-**

- t-if, t-elif, t-else
- t-out
- t-esc
- t-call

```
<t t-call="web.html_container">
    <t t-foreach="docs" t-as="o">
        <t t-call="web.external_layout">
            <div class="page">
                <h2>Report title</h2>
            </div>
        </t>
    </t>
</t>
```

# Qweb Template

**QWeb** is the primary **templating** engine used by Odoo

It is an XML templating engine and used mostly to generate HTML fragments and pages.

Template directives are specified as XML attributes prefixed with **t-**

- t-if, t-elif, t-else
- t-out
- t-esc
- t-call
- t-foreach, t-as

```
<t t-foreach="[1, 2, 3]" t-as="i">
   <p><t t-out="i"/></p>
</t>
```

Will be rendered as:

```
<p>1</p>
<p>2</p>
<p>3</p>
```

# Qweb Template

**QWeb** is the primary **templating** engine used by Odoo

It is an XML templating engine and used mostly to generate HTML fragments and pages.

Template directives are specified as XML attributes prefixed with **t-**

- t-if, t-elif, t-else
- t-out
- t-esc
- t-call
- t-foreach, t-as
- t-set, t-value

```
<t t-set="existing_variable" t-value="False"/>
<!-- existing_variable now False -->
```

# Practicals

**Create a module**

**Create a model**

**Define new menu entries**
Define new menu entries to access courses under the OpenAcademy menu entry. A user should be able to :

- display a list of all the courses
- create/modify courses

**Create a report for the Course model**
For each course, it should display its name

# Q&A