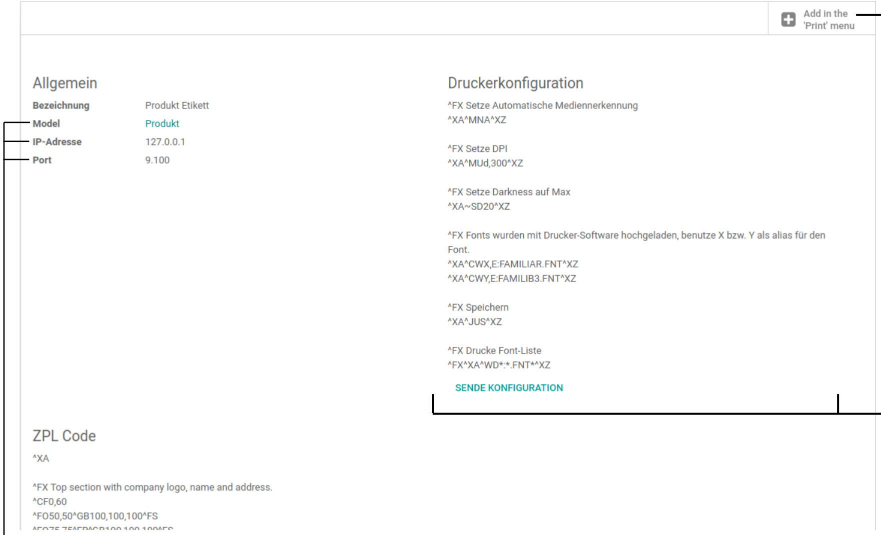
DATENPOL ZPL MODULE DOCUMENTATION

INSTALLATION GUIDE

1. Install the module „dp_print_zpl”
2. Click on “ZPL” in the settings menu
3. Configure dynamic labels and printers
4. Create necessary labels in Jinja 2
5. The ZPL printing functions are ready to use now

CREATING DYNAMIC LABELS AND PRINTERS



Allgemein

Bezeichnung	Produkt Etikett
Model	Produkt
IP-Adresse	127.0.0.1
Port	9.100

Druckerkonfiguration

```
^FX Setze Automatische Mediennerkennung
^XA^MNA^XZ

^FX Setze DPI
^XA^MUd,300^XZ

^FX Setze Darkness auf Max
^XA~SD20^XZ

^FX Fonts wurden mit Drucker-Software hochgeladen, benutze X bzw. Y als alias für den
Font.
^XA^CWX,E:FAMILIAR.FNT^XZ
^XA^CWY,E:FAMILIB3.FNT^XZ

^FX Speichern
^XA^JUS^XZ

^FX Drucke Font-Liste
^FX^XA^WD*:*.FNT*^XZ
```

ZPL Code

```
^XA

^FX Top section with company logo, name and address.
^CF0,60
^FO50,50^GB100,100,100^FS
^FO75,75^FR^GB100,100,100^FS
```

This button adds a “Print“ button to the printing menu of the defined model.

Enter the printer configuration and press “send”.

Each label can be configured to address a different printer.

TESTING ZPL LABELS

In order to get your ZPL Code right, it is best to run a few tests to see how the label looks. We recommend two ways to do this.

ONLINE ZPL VIEWER

http://labelary.com/viewer.html
Copy your desired code into the Viewer, and it will immediately render a view of how your label will look printed.

ZPL PRINTER CHROME APP/EXTENSION

https://chrome.google.com/webstore/detail/zpl-printer/phoidlklenidapnijkabnfdgmadlcmjo
Based on labelary, this Chrome browser extension allows you to preview and save your ZPL labels more efficiently.

USING JINJA2 TO CREATE LABELS

Jinja2 is a powerful markup engine that offers a huge range of possibilities when designing and managing labels. This allows you to more precisely control how the labels are compiled and create new use cases for the module in your enterprise.

LOOPS

Jinja2 offers the possibility of using loops in order to perform certain tasks more than once.

1.1 Example: Printing the same label 7 times:

```
{% for i in range(7) %}
## your template here
{% endfor %}
```

1.2 Example: Print all entries in a table

```
{% for quant in model.quant_ids%}
{{ quant.lot_id.name }}
 {% endfor %}
```

1.3 Example: Printing labels depending on Package size and Amount.

```
{% for quant in model.quant_ids -%}
{% for i in range((quant.qty *
quant.product_uom_id.factor)|int) %}
{{ quant.lot_id.name }}
{% endfor %} {% endfor %}
```

The function **range()** can only be called with integers. Since the amounts in Odoo are stored as floating point values, these need to be converted to integers first. This can be done using following function:

```
{% for i in range(quant.qty|int) %}
{% endfor %}
```

The **|int** after the floating point value is the important element.

QUERIES

Queries in Jinja2 can be used to display different values and data depending on query results. Example:

```
{% if model.lot_id %}
## First template
{% else %}
## Second template
{% endif %}
```

USAGE OF DATA

```
{{ model.name }}
```

Data can be used within a template in Jinja2 using following command

USAGE OF AUXILIARY VARIABLES

If an expression is used multiple times in one template, it can be saved within the template as a variable for later usage.

```
{% set quantity = 8 %}
```

FULL LABEL EXAMPLE

```
^FX Set label length
^XA^LL600^XZ
^FX Font was uploaded using zebra-designer (windows software), use Z as an alias
for the font.
^XA^CWZ,E:ARI001.FNT^XZ
^XA
^FX set encoding UTF-8
^CI28
^FX header box
^FO25,25^GB766,100,2,B,0^FS
^FX header box divider
^FO300,25^GB1,100,2,B,0^FS

^FX grid and left column
^FO25,25^GB766,558,2^FS
^FO25,300^GB766,1,2^FS
^FO25,520^GB766,1,2^FS
^CI27
^FX ad image
^FO40,30^FR
^GFA,1705,1705,31,,::::03hWFE,:03MF8hO0E,03MF8hO06,:::03MF8gG07F8I0FFg06
,:::::::03CK
078OFE03MF807IF8K0MFE03MFC6,::::::03CK078OFE03MF807IFL0MFE03MFC6,03C
03JF8F
F01FE01FE03FCI07F807F8I0FF00FFI01FE03FCI03FC6,::::::03CK078FF01FE01FE03MF
807IF0
0FF00MFE03FCI03FC6,03CK078FF01FE01FE03MF807IF80FF00MFE03FCI03FC6,::::::
03MF8
hO06,::::03hWFE,::,:::^FS
^FS

^CFZ,32
^FO80,85^FDService^FS
^FO81,85^FDService^FS
^FO82,85^FDService^FS
^FO83,85^FDService^FS
```

FULL LABEL EXAMPLE (CONT.)

```
^FX Date check - in in font size 32
^CFZ,32
^FO320,60^FD{{ model.confirmation_date }}^FS

^FX Company name, font size 24, centered
^CFZ,32
^FO35,150
^FB750,1,,C^FD{{model.partner_id.name}}^FS

^FX Number, font size 24, centered
^CFZ,80
^FO40,205
^FB760,1,,C^FD{{ model.name }}^FS

^FX Number, font size 24, centered
^CFZ,24
^FO50,320
^FDIn warranty^FS
^FO50,390
^FDPartially in warranty^FS
^FO50,460
^FDNo warranty^FS

^FO500,374^FDKV created^FS
^FO460,370^GB25,1,2^FS
^FO460,370^GB1,25,2^FS
^FO460,395^GB25,1,2^FS
^FO485,370^GB1,25,2^FS

^FH^FO500,455^FDKV confirmed^FS
^FO460,450^GB25,1,2^FS
^FO460,450^GB1,25,2^FS
^FO460,475^GB25,1,2^FS
^FO485,450^GB1,25,2^FS
^FX "www.datenpol.at"
^CFZ,24
^FO40,540^FD{{model.responsible_completion.name}}^FS
^XZ
```