



User manual to define a patient form



TO DEFINED A PATIENT FORM

[form_patient_lang.toml](#) (UTF-8 encoding)

Your toml file must follow the naming convention, `form_patient_LANG.toml`, where LANG is a language available in LabBook (fr, uk, us, es, ar, km, lo, mg, pt). To test your file via the preview in the administrator area, you can upload your file with any ending you like. It should start with "form_patient_" and end with ".toml". Of course, if you're editing in several languages, the descriptions must be the same, otherwise you run the risk of having too little or too much information depending on the language you're editing in.

- A new toml file can be submitted and tested via the menu Settings => Configuring forms
- By default `form_patient_fr.toml` will be used even in another language selected if the other file doesn't exist
- Tabulations are important
- 2 mandatory sections: [description] and [layout]

Advice: It's recommended that you run your modified toml through a validator such as <https://www.toml-lint.com/>

Description

[description] includes the description of the elements to be positioned in the layout section.

A `form_element` block begins with : `[[description.form_element]]` Below this block start, several characteristics can be defined

2 types of elements, predefined and unknown.

List of predefined element :

pat_ano : choose anonymous, Yes or No [input_type = "radio"]

pat_code_lab : enter a string [input_type = "text"] **pat_code** : display a internal unique code associated to this patient

pat_name : enter name of patient [input_type = "text"]

pat_midname : enter midname of patient [input_type = "text"]

pat_maiden : enter maiden of patient [input_type = "text"]

pat_firstname : enter firstname of patient [input_type = "text"]

pat_sex : choose patient sex, Male, Female or Unknown [input_type = "radio"]

pat_birth : pick a date of birth for this patient [input_type = "select"]

pat_birth_approx : choose if the patient date of birth is approximative, Yes or No [input_type = "radio"]

pat_age : enter age of patient [input_type = "number"] **pat_age_unit** : choose a age unit, Day, Week, Month, Year [input_type = "select"]

pat_nationality : choose a nationality in a list [input_type = "select"]

pat_resident : choose if this patient is a resident, Yes or No [input_type = "radio"]

pat_blood_group : choose blood group of this patient, A, B, AB, O [input_type = "select"]

pat_blood_rhesus : choose blood rhesus of this patient, + or - [input_type = "select"]

pat_address : enter an address of patient [input_type = "textarea"]

pat_phone1 : enter a phone number [input_type = "text"]

pat_phone2 : enter a phone number [input_type = "text"]

pat_profession : enter profession of patient [input_type = "text"]

search_zipcity : type some number or letter to find zipcode and city, works if pat_zipcode or/and pat_city are used

pat_pbox : enter a postal box [input_type = "text"] **pat_district** : enter a district [input_type = "text"]

pat_zipcode : enter a zipcode [input_type = "text"]

pat_city : enter a city name [input_type = "text"]

List of possible features for predefined element (bold features are mandatory) : **labbook_ref** = "*name of predefinedelement*" **input_type** = "*input type corresponding to the reference*" [Note : excepted for pat_code and search_city]

List of possible features for unknow element (bold features are mandatory) : **id** = "*unique name of id*" **label** = "*element label displayed*" **type** = "*type of element*" (useful for displayed title...)

list of type : h1, h2, h3, h4, h5, h6, span

OR

input_type = "*type of element*" (useful for entered or choosed data)

list of input_type : text, textarea, number, select, radio [Note : The last 2 require options to be defined]

options = [{ value = "", label = ""}, { value = "", label = "" }, ...] [Note : mandatory for input_type radio or select]

List of optionnals attributes : attr_required = true [Note : Make field mandatory and add auto * after label] attr_value = "default text or number ..."

[Note : useful for input type textarea] attr_rows = "4" attr_cols = "50"

[Note : useful for input type number] attr_min = "0" attr_max = "10" attr_step = "1"

Examples of custom elements:

```
# TEST 01 : select choice between no answer, OK and KO
[[description.form_element]]
id = "test_01"
label = "Test 01"
input_type = "select"
options = [ { value = "", label = "" }, { value = "OK", label = "OK" }, { value = "KO", label = "KO" } ]

# TEST 02 : datetime-local
[[description.form_element]]
id = "test_02"
label = "Test 02"
input_type = "datetime-local"

# TEST 03 : number between 0 and 10 by step 1
[[description.form_element]]
id = "test_03"
label = "Test 03"
input_type = "number"
attr_min = "0"
attr_max = "10"
attr_step = "1"

# TEST 04 : required text
[[description.form_element]]
id = "test_04"
label = "Test 04"
input_type = "text"
attr_required = true

# TEST 05 : custom radio
[[description.form_element]]
id = "test_06"
label = "Test 06"
input_type = "radio"
options = [
  { value = "", label = "Unknown" },
```



```
{ value = "OK", label = "OK" },
{ value = "KO", label = "KO" }
]

# TEST 06 : textarea
[[description.form_element]]
id = "test_07"
label = "Test 07"
input_type = "textarea"
attr_value = "TEST Textarea"
attr_rows = "4"
attr_cols = "50"
```

layout

[layout] includes a description of the layout of the elements described above

A row begins with : [[layout.rows]]

class = "list of class"

class examples : row : define a row col : define a column mt-x : define a margin top with x = 1 smaller margin to 3 bigger margin, possible to replace t for top, by b for bottom, s for start, e for end, x for horizontal, y for vertical. panel-heading : horizontal line below panel-title : green and bold font flex-row : force to align elements horizontally flex-md-row : force to align elements horizontally if viewport are bigger >= 768px (see <https://getbootstrap.com/docs/5.0/layout/breakpoints/>) flex-col : force to align elements vertically

For more classes, see the bootstrap 5 documentation. <https://getbootstrap.com/docs/5.0/getting-started/introduction/>

In a row, you can define another container element (row, column) or start arranging one or more elements.

Some examples :

```
[[layout.rows]]
class = "panel-heading row"
elements = [ { element = "label_identity", class = "panel-title" } ]
```

Position the identity title on this line

```
[[layout.rows.cols]]
class = "col-md-7"
```

```
elements = [ { element = "pat_code_lab", class = "flex-md-row" } ]
```

Places the entry of a laboratory code for the patient in this column, which takes up a lot of space if the screen is wide enough.

```
[[layout.rows.cols]]
class = "col-md"
elements = [
  { element = "pat_name", class = "flex-md-row" },
  { element = "pat_midname", class = "flex-md-row" },
  { element = "pat_maiden", class = "flex-md-row" },
  { element = "pat_firstname", class = "flex-md-row" },
  { element = "pat_sex", class = "flex-md-row" }
]
```

Position several elements in this column, each of which will be placed on a separate line.

```
[[layout.rows.cols]]
class = "col-md"
elements = [
  { element = "pat_birth", class = "flex-md-row" },
  { element = "pat_birth_approx", class = "flex-md-row" },
  { class = "d-md-flex", elements = [
    { element = "label_or", class = "flex-md-row" },
    { element = "pat_age", class = "flex-md-row" },
    { element = "pat_age_unit", class = "flex-md-row" }
  ]},
  { element = "pat_nationality", class = "flex-md-row" },
  { element = "pat_resident", class = "flex-md-row" },
  { class = 'd-md-flex', elements = [
    { element = "pat_blood_group", class = "flex-md-row" },
    { element = "pat_blood_rhesus", class = "flex-md-row" }
  ]}
]
```

Position several elements in this column, some of which will be grouped together on the same line. d-md-flex is used to define a flexbox container and transform direct child's elements into flex items. Flex containers and items can be modified further with additional flex properties.

