



Fondation
Mérieux

Lab | Book

LabBook 3.6

Define patient file and patient history form

V2

January 2026

Fondation Mérieux

Lutte contre les maladies infectieuses depuis 1967

www.fondation-merieux.org



Table of contents

1.	PATIENT FORM SPECIFICATION	2
1.1.	Description	2
1.2.	Page layout.....	4
2.	PATIENT HISTORY FORM SPECIFICATIONS	6
2.1.	File location and naming.....	6
2.2.	Mandatory sections	7
2.3.	Naming rules for block fields.....	9

1. PATIENT FILE SPECIFICATION

The 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. It must begin 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 little or a lot of information depending on the editing language.

A new toml file can be submitted and tested via the menu Parameters => Form configuration

By default, form_patient_en.toml will be used even in another selected language if the other file does not exist. Tabs are important.

2 sections are mandatory: [description] and [layout].

Tip: We recommend testing your toml file with a validator such as <https://www.toml-lint.com/>.

1.1. Description

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

A form_element block starts with : [[description.form _element]] Below this block begins, several characteristics can be defined.

There are 2 types of element, predefined and unknown.

List of predefined elements

Field	Name / Description	Type of input	Options / Remarks
pat_ano	Anonymous patient	Radio	Yes / No
pat_code_lab	Internal patient code laboratory	Text	Free string
pat_code	Unique internal patient code	Display	Automatically generated
pat_name	Patient name	Text	-
pat_midname	Patient's middle name patient	Text	-
pat_maiden	Patient's maiden name patient	Text	-
pat_firstname	Patient's first name	Text	-
pat_sex	Patient's gender	Radio	Male / Female / Unknown
pat_birth	Date of birth of patient	Select	-
pat_birth_approx	Date of birth approximate	Radio	Yes / No
pat_age	Patient's age	Number	-
pat_age_unit	Age unit	Select	Day / Week / Month / Year
pat_nationality	Patient's nationality	Select	Predefined list

pat_resident	Resident patient	Radio	Yes / No
pat_blood_group	Blood group	Select	A / B / AB / O
pat_blood_rhesus	Blood rhesus	Select	+ / -
pat_address	Patient address	Textarea	-
pat_phone1	Main telephone main	Text	-
pat_phone2	Secondary phone secondary	Text	-
pat_profession	Patient's profession	Text	-
search_zipcity	Search postal code / city	Search	Works if pat_zipcode and/or pat_city are used
pat_pbox	Postal box	Text	-
pat_district	District	Text	-
pat_zipcode	Zip code	Text	-
pat_city	City	Text	-

List of possible functionalities for predefined elements (bold characters are mandatory)

labbook_ref = "name of predefined element" input_type

= "input type corresponding to reference" [Note: except pat_code and search_city].

List of possible functionalities for the unknown element (bold characters are mandatory): id = "unique id name" label = "displayed element label" type = "element type" (useful for displaying title ...)

List of types: h1, h2, h3, h4, h5, h6, span OR

input_type = "element type" (useful for input or selected 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 or select radio element]

List of attribute options: attr_required = true [Note: Makes field mandatory and adds 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: selects between no response, 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"
```

1.2. Layout

[layout] includes a description of the layout of the elements described above a row begins with :
[[layout.rows]]

class = "class list

Examples of classes :

row: defines a row col:

defines a column

mt-x: defines an upper margin with x = 1 smallest margin to 3 largest margin, t for top can be replaced by b for bottom, s for start, e for end, x for horizontal, y for vertical.

panel-heading: horizontal line below panel-title: bold, green text font flex-row: forces elements to be aligned horizontally

flex-md-row: forces elements to align horizontally if window size >= 768px (see <https://getbootstrap.com/docs/5.0/layout/breakpoints/>).

For more information, see 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 organizing one or more elements.

Here are a few examples:

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

Position the title identity on this row.

```
[[layout.rows.cols]]
class = "col-md-7"
elements = [ { element = "pat_code_lab", class = "flex-md-row" } ]
```

Places the entry of a patient's laboratory code 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 elements into flexible elements. Containers and flex elements can be further modified with additional flex properties.

2. PATIENT HISTORY FORM SPECIFICATION

This section describes how to define and configure patient history forms in LabBook using TOML files.

History forms allow the user to enter, modify and view patient history information, grouped in blocks.

All files must be UTF-8 encoded.

2.1. File location and naming **Location :**

history forms are stored in: `storage/resource/form/patient/`

Naming convention :

`form_patient_hist_LANG.toml`

LANG must correspond to a language code recognized by LabBook (fr, uk, us, es, ar, km, lo, mg, pt).

Examples:

- form_patient_hist_fr.toml
- form_patient_hist_en.toml

Any test file can be uploaded via the administrator interface, provided it begins with form_patient_hist_ and ends with .toml.

2.2. Mandatory sections

A history form TOML file must contain at least the following three t o p - l e v e l sections:

[description]

[history]

[layout]

No other first-level sections are allowed.

Section: description

This section contains all the fields that may appear in one or more history blocks.

Each field is declared with :

[[description.form_element]]

There are two categories of fields.

Custom fields (most common)

Mandatory keys :

- Id: unique identifier of the element
- Label: label displayed to the user
- input_type - type of component (widget)

Possible values for input_type :

- text
- textarea
- number

- select
- radio
- date
- datetime-local

For radio and select types, it is mandatory to define:

```
options = [
  { value = "X", label = "Text displayed" },
  ...
]
```

Optional attributes :

attr_required = true # mandatory field, a "*" is added after the label

attr_value = "default content"

for textarea

attr_rows = "4"

attr_cols = "50"

for number

attr_min = "0"

attr_max = "10"

attr_step = "1"

Display elements only

```
[[description.form_element]]
```

```
id = "my_title"
```

```
label = "My title" type
```

```
= "h3"
```

Allowed types: h1, h2, h3, h4, h5, h6, span.

Section : history

This section defines :

- how the form history is structured in blocks
- fields visible as table columns

Example:

```
[history]
[[history.block]]
id = "clin"
label = "Clinical signs"
columns = ["clin_signs", "clin_datetime"]
```

Rules:

- id is the prefix of all fields belonging to this block Example:
block clin → fields starting with clin_
- columns lists the subset of fields visible in the table
The other fields still exist and can be modified, but are not displayed in the table.

Several blocks can be defined:

```
[history]
[[history.block]]
id = "clin"
columns = ["clin_signs", "clin_datetime"]
[[history.block]]
id = "treat"
columns = ["treat_drug", "treat_comment"]
```

2.3. Block field naming rule

Any field belonging to a block must begin with: block_id + "_"

Example:

```
[[description.form_element]]
id = "clin_signs"
label = "Clinical signs"
input_type = "text"
```

```
[[description.form_element]]
id = "clin_datetime"
label = "Date and time"
input_type = "datetime-local"
```

- Block identifier : clin
- Field names : clin_signs, clin_datetime

Minimal functional example: patient history form

```
# =====
# Minimal patient history form
# Example file: form_patient_hist_en.toml
# =====

[description]

# "med" block: very simple medical history #

Title (cosmetics only)
[[description.form_element]]
id = "label_med"
label = "Medical history" type =
"h3"

# Date and time of event [[description.form_element]]
id = "med_datetime"
label = "Date and time"
input_type = "datetime-local"

# Comment
[[description.form_element]]
id = "med_comment"
label = "Comment"
input_type = "textarea"
attr_rows = "3"
attr_cols = "40"

# =====
# SECTION LAYOUT (mandatory)
# =====

[layout]

# Block title
[[layout.rows]]
class = "panel-heading row"
elements = [
  { element = "label_med", class = "panel-title" }
]

# Form fields + history table [[layout.rows]]
class = "row mt-2"
elements = [
  # Form on left
  { class = "col-md-6", elements = [
    { element = "med_datetime", class = "flex-md-row" },
    { element = "med_comment", class = "flex-md-row" }
  ] },
  # History table on right (automatically generated for "med" block)
  { element = "med", class = "col-md-6" }
]

# =====
```

```
# HISTORY SECTION (mandatory for history forms)
# =====

[history]

[[history.block]]
id    = "med"
columns = [
    "med_datetime",
    "med_comment"
]
```

Fondation Mérieux

Fighting infectious diseases since 1967

www.fondation-merieux.org