



Fondation
Mérieux

Lab | Book

LabBook 3.6

Définir fiche patient et un formulaire d'historique du patient

V2

Janvier 2026

Fondation **Mérieux**

Lutte contre les maladies infectieuses depuis 1967

www.fondation-merieux.org



Table des matières

1.	SPECIFICATION DE LA FICHE PATIENT	2
1.1.	Description.....	2
1.2.	Mise en page.....	4
2.	SPECIFICATION DU FORMULAIRE HISTORIQUE PATIENT	6
2.1.	Emplacement et nommage des fichiers	6
2.2.	Sections obligatoires.....	7
2.3.	Règle de nommage des champs d'un bloc	9

1. SPECIFICATION DE LA FICHE PATIENT

Le fichier toml doit suivre la convention de dénomination, **form_patient_LANG.toml**, où LANG est un langage disponible en LabBook (fr , uk , us , es , ar , km , lo , mg , pt). Pour tester votre fichier via l'aperçu dans l'espace administrateur. Il doit commencer par « form_patient _ » et se terminer par « . toml ». Bien sûr, si vous éditez dans plusieurs langues, les descriptions doivent être les mêmes, sinon vous courez le risque d'avoir peu ou beaucoup d'informations selon la langue d'édition.

Un nouveau fichier toml peut être soumis et testé via le menu Paramètres => Configuration des formulaires

Par défaut form_patient_ fr.toml sera utilisé même dans une autre langue sélectionnée si l'autre fichier n'existe pas. Les tabulations sont importantes.

2 sections sont obligatoires : [description] et [layout]

Conseil : Il est recommandé de tester votre fichier toml élaboré via un validateur tel que <https://www.toml-lint.com/>.

1.1. Description

[description] comprend la description des éléments à positionner dans la section layout

Un bloc form_element commence avec : [[description.form _element]] Ci-dessous ce bloc commence, plusieurs caractéristiques peuvent être définies.

Il y'a 2 types d'éléments, prédéfinis et inconnus.

Liste des éléments prédéfinis

Champ	Libellé / Description	Type de saisie	Options / Remarques
pat_ano	Patient anonyme	Radio	Oui / Non
pat_code_lab	Code patient interne au laboratoire	Text	Chaîne libre
pat_code	Code interne unique du patient	Affichage	Généré automatiquement
pat_name	Nom du patient	Text	—
pat_midname	Deuxième prénom du patient	Text	—
pat_maiden	Nom de jeune fille du patient	Text	—
pat_firstname	Prénom du patient	Text	—
pat_sex	Sexe du patient	Radio	Homme / Femme / Inconnu
pat_birth	Date de naissance du patient	Select	—
pat_birth_approx	Date de naissance approximative	Radio	Oui / Non
pat_age	Âge du patient	Number	—
pat_age_unit	Unité de l'âge	Select	Jour / Semaine / Mois / Année
pat_nationality	Nationalité du patient	Select	Liste prédéfinie

pat_resident	Patient résident	Radio	Oui / Non
pat_blood_group	Groupe sanguin	Select	A / B / AB / O
pat_blood_rhesus	Rhésus sanguin	Select	+ / -
pat_address	Adresse du patient	Textarea	—
pat_phone1	Numéro de téléphone principal	Text	—
pat_phone2	Numéro de téléphone secondaire	Text	—
pat_profession	Profession du patient	Text	—
search_zipcity	Recherche code postal / ville	Recherche	Fonctionne si pat_zipcode et/ou pat_city sont utilisés
pat_pbox	Boîte postale	Text	—
pat_district	Quartier	Text	—
pat_zipcode	Code postal	Text	—
pat_city	Ville	Text	—

Liste des fonctionnalités possibles pour les éléments prédéfinis (les caractères gras sont obligatoires)
: labbook_ref = " nom de l'élément prédéfini " input_type

= " type d'entrée correspondant à la référence " [Remarque : sauf pat_code et search_city]

Liste des fonctionnalités possibles pour l'élément inconnu (les caractères gras sont obligatoires) : id = " nom unique de l'id " label = " libellé de l'élément affichée " type = " type d'élément " (utile pour afficher titre ...)

Liste des types : h1, h2, h3, h4, h5, h6, span OU

input_type = " type d'élément " (utile pour les données saisies ou choisies)

Liste de input_type : text , textarea , number , select, radio [Remarque : les 2 dernières nécessitent que les options soient défini]

options = [{ value = "", label = "" }, { value = "", label = "" }, ...] [Remarque : obligatoire pour l'élément radio input_type ou select]

Liste des options attributs : attr_required = true [Remarque : Rend le champ obligatoire et ajoute auto * après label] attr_value = " texte ou numéro par défaut ... "

[Remarque : utile pour le type d'entrée textarea] attr_rows = "4" attr_cols = "50"

[Remarque : utile pour le numéro de type d'entrée] attr_min = "0" attr_max = "10" attr_step = "1"

Exemples d'éléments personnalisés

TEST 01 : sélectionne le choix entre pas de réponse, OK et 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. Mise en page

[layout] inclut une description de la disposition des éléments décrit au-dessus d'une rangée commence avec : [[layout.rows]]

class = "liste des classes"

Exemples de classe :

row : définit une ligne

col : définit une colonne

mt-x : définit une marge supérieure avec x = 1 plus petite marge à 3 plus grande marge, possible de remplacer t pour haut, par b pour bas, s pour début, e pour fin, x pour horizontal, y pour vertical.

panel-heading : ligne horizontale en dessous panel-title : police du texte gras et verte

flex-row : force à aligner les éléments horizontalement

flex-md-row : force à aligner les éléments horizontalement si la fenêtre est plus grande ≥ 768 px (voir plus <https://getbootstrap.com/docs/5.0/layout/breakpoints/>).

Pour en savoir plus, consultez la documentation Bootstrap 5.

<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

Dans une rangée , vous pouvez définir un autre élément conteneur (ligne , colonne) ou commencez à organiser un ou plusieurs éléments.

Quelques exemples :

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

Positionner l'identité titre sur cette ligne.

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

Place la saisie d'un code laboratoire du patient dans cette colonne, ce qui prend beaucoup de place si l'écran est suffisamment large.

```
[[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" }
]
```

Positionnez plusieurs éléments dans cette colonne, chacun d'eux sera placé sur une ligne distincte.

```
[[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" }
  ] }
]
```

Positionnez plusieurs éléments dans cette colonne dont certains seront regroupés sur une même ligne. `d-md-flex` est utilisé pour définir un conteneur flexbox et transformer les éléments enfants directs en éléments flexibles. Les conteneurs et éléments flexibles peuvent être modifiés davantage avec des propriétés flexibles supplémentaires.

2. SPECIFICATION DU FORMULAIRE HISTORIQUE PATIENT

Cette partie décrit comment définir et configurer des formulaires d'historique patient dans LabBook à l'aide de fichiers TOML.

Les formulaires d'historique permettent à l'utilisateur de saisir, modifier et consulter des informations historiques du patient, regroupées par blocs.

Tous les fichiers doivent être encodés en UTF-8.

2.1. Emplacement et nommage des fichiers

Emplacement :

les formulaires d'historique sont stockés dans : `storage/resource/form/patient/`

Convention de nommage :

`form_patient_hist_LANG.toml`

LANG doit correspondre à un code langue reconnu par LabBook (fr, uk, us, es, ar, km, lo, mg, pt).

Exemples :

- form_patient_hist_fr.toml
- form_patient_hist_en.toml

Il est possible de téléverser n'importe quel fichier de test via l'interface administrateur, à condition qu'il commence par form_patient_hist_ et se termine par .toml.

2.2. Sections obligatoires

Un fichier TOML de formulaire d'historique doit contenir au minimum les trois sections de premier niveau suivantes :

[description]

[history]

[layout]

Aucune autre section de premier niveau n'est autorisée.

Section : description

Cette section contient l'ensemble des champs pouvant apparaître dans un ou plusieurs blocs d'historique.

Chaque champ est déclaré avec :

[[description.form_element]]

Il existe deux catégories de champs.

Champs personnalisés (les plus courants)

Clés obligatoires :

- Id : identifiant unique de l'élément
- Label : libellé affiché à l'utilisateur
- input_type — type de composant (widget)

Valeurs possibles pour input_type :

- text
- textarea
- number

- select
- radio
- date
- datetime-local

Pour les types radio et select, il est obligatoire de définir :

```
options = [
  { value = "X", label = "Texte affiché" },
  ...
]
```

Attributs optionnels :

`attr_required = true` # champ obligatoire, un "*" est ajouté après le libellé

`attr_value` = "contenu par défaut"

pour textarea

`attr_rows` = "4"

`attr_cols` = "50"

pour number

`attr_min` = "0"

`attr_max` = "10"

`attr_step` = "1"

Éléments d'affichage uniquement

`[[description.form_element]]`

`id` = "my_title"

`label` = "Mon titre"

`type` = "h3"

Types autorisés : h1, h2, h3, h4, h5, h6, span.

Section : history

Cette section définit :

- la manière dont l'historique du formulaire est structuré en blocs
- les champs visibles comme colonnes du tableau

Exemple :

```
[history]
[[history.block]]
id = "clin"
label = "Signes cliniques"
columns = ["clin_signs", "clin_datetime"]
```

Règles :

- id est le préfixe de tous les champs appartenant à ce bloc
Exemple : bloc clin → champs commençant par clin_
- columns liste le sous-ensemble de champs visibles dans le tableau
Les autres champs existent toujours et sont modifiables, mais non affichés dans le tableau.

Il est possible de définir plusieurs blocs :

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

2.3. Règle de nommage des champs d'un bloc

Tout champ appartenant à un bloc doit commencer par :

block_id + "_"

Exemple :

```
[[description.form_element]]
id = "clin_signs"
label = "Signes cliniques"
input_type = "text"
```

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

- Identifiant du bloc : clin
- Noms des champs : clin_signs, clin_datetime

Exemple minimal fonctionnel : formulaire d'historique patient

```
# =====
# Formulaire minimal d'historique patient
# Exemple de fichier : form_patient_hist_fr.toml
# =====

[description]

# Bloc "med" : historique médical très simple

# Titre (cosmétique uniquement)
[[description.form_element]]
id   = "label_med"
label = "Antécédents médicaux"
type = "h3"

# Date et heure de l'événement
[[description.form_element]]
id       = "med_datetime"
label    = "Date et heure"
input_type = "datetime-local"

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

# =====
# SECTION LAYOUT (obligatoire)
# =====

[layout]

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

# Champs du formulaire + tableau d'historique
[[layout.rows]]
class = "row mt-2"
elements = [
  # Formulaire à gauche
  { class = "col-md-6", elements = [
    { element = "med_datetime", class = "flex-md-row" },
    { element = "med_comment", class = "flex-md-row" }
  ] },
  # Tableau d'historique à droite (généré automatiquement pour le bloc "med")
  { element = "med", class = "col-md-6" }
]

# =====
```

```
# SECTION HISTORY (obligatoire pour les formulaires d'historique)
# =====

[history]

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



Fondation **Mérieux**

Lutte contre les maladies infectieuses depuis 1967

www.fondation-merieux.org

