



Définir une fiche patient

LabBook Version 3.4

POUR DÉFINIR UNE FICHE PATIENT

Ton 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/>.

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.

2 types d'éléments, prédéfinis et inconnus

Liste des éléments prédéfinis : pat_ano : choisir anonyme , Oui ou Non [input_type = "radio"] pat_code_lab : saisir une chaîne [input_type = " text "] pat_code : afficher un code interne unique associé à ce patient
pat_name : saisir le nom du patient [input_type = " text "] pat_midname : entrez le prénom du patient [input_type = " text "] pat_maiden : entrez la jeune fille du patient [input_type = " text "] pat_firstname : entrez le prénom du patient [input_type = " text "] pat_sex : choisissez le sexe du patient , Homme, Femme ou Inconnu [input_type = "radio"] pat_birth : choisit une date de naissance pour ce patient [input_type = "select"] pat_birth_approx : choisit si la date de naissance du patient est approximatif, Oui ou Non [input_type = "radio"] pat_age : entrez l'âge du patient [input_type = " number "] pat_age_unit : choisissez une unité d'âge , Jour, Semaine, Mois , Année [input_type = "select"] pat_nationality : choisissez une nationalité dans une liste [input_type = "select"] pat_resident : choisissez si ce patient est résident ,

Oui ou Non [input_type = " radio "] pat_blood_group : choisissez groupe sanguin de ce patient, A, B, AB, O [input_type = "select"] pat_blood_rhesus : choisir sang rhésus de ce patient, + ou - [input_type = "select"] pat_address : saisissez une adresse du patient [input_type = " textarea "] pat_phone1 : saisissez un numéro de téléphone [input_type = " text "] pat_phone2 : saisissez un numéro de téléphone [input_type = " text "] pat_profession : saisissez la profession du patient [input_type = " text "] search_zipcity : saisissez quelques chiffre ou lettre à trouver zipcode et city, fonctionne si pat_zipcode ou/et pat_city sont utilisés pat_pbox : saisissez une boîte postale [input_type = " text "] pat_district : saisissez un quartier [input_type = " text "] pat_zipcode : saisissez un code postal [input_type = " text "] pat_city : saisissez un nom de ville [input_type = " 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" }, { valeur = "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"

```

Mise en page

[layout] inclut une description de la disposition des éléments décrit au-dessus de

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 $\geq 768\text{px}$ (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.