**FormBodyTabbed Illustration**
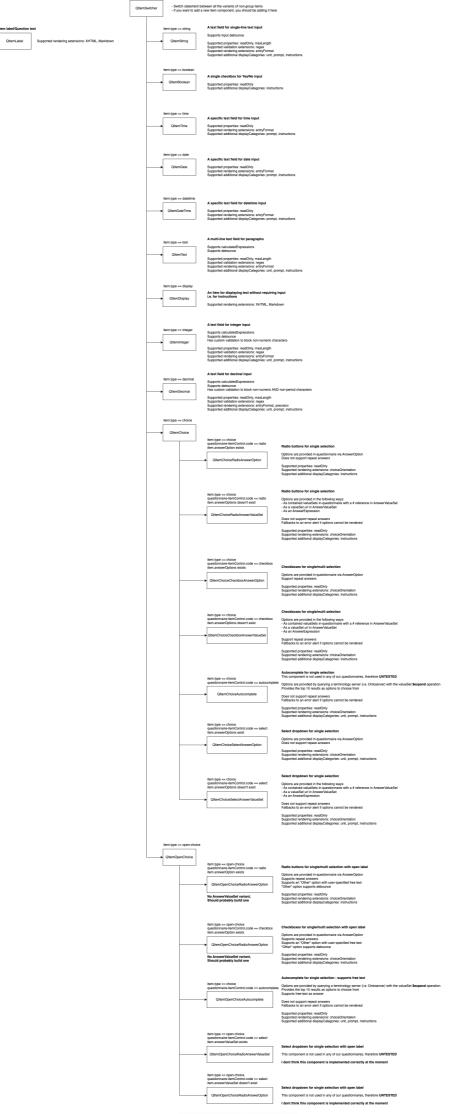
| TabList | QItemGroup |
|---|---|
| Tab1 | |
| Tab2 | |
| ... | |

**RendererLayout**
- Initialise QR upon new form creation
- Contains state of QR and current tab (if there are tabs)
- Contains population logic
- Contains code section to manage Q changes from server
  (code not ideal atm, need to rewrite this)
- Contains nav bar (preview, save functions etc)
- Contains route-blocker to prevent user from exiting form when there are unsaved changes

Connected via React-router

**Form**
- Contains state for preprocessed (aka cached) valueset codings
- Set values for EnableWhen upon form initialisation
- Update QR on form changes - also run fhirpath evaluation to update calculated expressions

If there are tabs

- If there are not tabs
- If top-level item type == group

- If there are not tabs
- If top-level item type != group

**FormBodyTabbed.tsx**
- Render a tab list
- Renders a tab container (which contains the visible QItemGroup)
- Contains state of tab indexes and whether they are marked as complete

**QItemGroup**
- Contains state of group and its answers
- if group is a tab, contains tab logic and render accordingly
- if group has repeats property, render accordingly
- Handle changes within the group and propagate changes upwards
  - changes are handled differently based on whether group has repeats property

**QItemSwitcher**
- Switch statement between all the variants of non-group items
- If you want to add a new item component, you should be adding it here

**QItemGroup**

contains other items...

**iterate each item in this group**

if QR item is an array
(this means that there are multiple answers)

if QR item is not an array
(this means that the item does not have answers,
or only one answer)

if itemControl == gtable

if itemControl != gtable

if item has repeat property

if item doesn't have repeat property

**QItemGroupTable**

Group table repeat group,
repeat items are rendered as rows

**QItemRepeatGroup**

Generic repeat group

if item.type == group

if item.type != group

if item.type == group

if item.type != group

if itemControl == gtable

if itemControl != gtable

**QItemGroupTable**

Group table repeat group,
repeat items are rendered as rows

**QItemRepeatGroup**

Generic repeat group

**QItemRepeat**

Generic repeat item

**QItemGroup**

**QItemSwitcher**

**Note: all items i.e. questionnaire.item OR item.item supports the questionnaire-hidden extension**

**QItemSwitcher**
- Switch statement between all the variants of non-group items
- If you want to add a new Item component, you should be adding it here

**Item label/Question text**

**QItemLabel** — Supported rendering extensions: XHTML, Markdown

item.type == string
**QItemString**

**A text field for single-line text input**

Supports input debounce

Supported properties: readOnly, maxLength
Supported validation extensions: regex
Supported rendering extensions: entryFormat
Supported additional displayCategories: unit, prompt, instructions

item.type == boolean
**QItemBoolean**

**A single checkbox for Yes/No input**

Supported properties: readOnly
Supported additional displayCategories: instructions

item.type == time
**QItemTime**

**A specific text field for time input**

Supported properties: readOnly
Supported rendering extensions: entryFormat
Supported additional displayCategories: prompt, instructions

item.type == date
**QItemDate**

**A specific text field for date input**

Supported properties: readOnly
Supported rendering extensions: entryFormat
Supported additional displayCategories: prompt, instructions

item.type == datetime
**QItemDateTime**

**A specific text field for datetime input**

Supported properties: readOnly
Supported rendering extensions: entryFormat
Supported additional displayCategories: prompt, instructions

item.type == text
**QItemText**

**A multi-line text field for paragraphs**

Supports calculatedExpressions
Supports debounce

Supported properties: readOnly, maxLength
Supported validation extensions: regex
Supported rendering extensions: entryFormat
Supported additional displayCategories: unit, prompt, instructions

item.type == display
**QItemDisplay**

**An item for displaying text without requiring input
i.e. for instructions**

Supported rendering extensions: XHTML, Markdown

item.type == integer
**QItemInteger**

**A text field for integer input**

Supports calculatedExpressions
Supports debounce
Has custom validation to block non-numeric characters

Supported properties: readOnly, maxLength
Supported validation extensions: regex
Supported rendering extensions: entryFormat
Supported additional displayCategories: unit, prompt, instructions

item.type == decimal
**QItemDecimal**

**A text field for decimal input**

Supports calculatedExpressions
Supports debounce
Has custom validation to block non-numeric AND non-period characters

Supported properties: readOnly, maxLength
Supported validation extensions: regex
Supported rendering extensions: entryFormat, precision
Supported additional displayCategories: unit, prompt, instructions

item.type == choice
**QItemChoice**

item.type == choice
questionnaire-itemControl.code == radio
item.answerOption exists
**QItemChoiceRadioAnswerOption**

**Radio buttons for single selection**

Options are provided in-questionnaire via AnswerOption
Does not support repeat answers

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: instructions

item.type == choice
questionnaire-itemControl.code == radio
item.answerOptions doesn't exist
**QItemChoiceRadioAnswerValueSet**

**Radio buttons for single selection**

Options are provided in the following ways:
- As contained valueSets in-questionnaire with a # reference in AnswerValueSet
- As a valueSet url in AnswerValueSet
- As an AnswerExpression

Does not support repeat answers
Fallbacks to an error alert if options cannot be rendered

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: instructions

item.type == choice
questionnaire-itemControl.code == checkbox
item.answerOptions exists
**QItemChoiceCheckboxAnswerOption**

**Checkboxes for single/multi selection**

Options are provided in-questionnaire via AnswerOption
Support repeat answers

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: instructions

item.type == choice
questionnaire-itemControl.code == checkbox
item.answerOptions doesn't exist
**QItemChoiceCheckboxAnswerValueSet**

**Checkboxes for single/multi selection**

Options are provided in the following ways:
- As contained valueSets in-questionnaire with a # reference in AnswerValueSet
- As a valueSet url in AnswerValueSet
- As an AnswerExpression

Support repeat answers
Fallbacks to an error alert if options cannot be rendered

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: instructions

item.type == choice
questionnaire-itemControl.code == autocomplete
**QItemChoiceAutocomplete**

**Autocomplete for single selection**
This component is not used in any of our questionnaires, therefore **UNTESTED**

Options are provided by querying a terminology server (i.e. Ontoserver) with the valueSet **Sexpand** operation
Provides the top 10 results as options to choose from

Does not support repeat answers
Fallbacks to an error alert if options cannot be rendered

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: unit, prompt, instructions

item.type == choice
questionnaire-itemControl.code == select
item.answerOptions exist
**QItemChoiceSelectAnswerOption**

**Select dropdown for single selection**

Options are provided in-questionnaire via AnswerOption
Does not support repeat answers

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: unit, prompt, instructions

item.type == choice
questionnaire-itemControl.code == select
item.answerOptions doesn't exist
**QItemChoiceSelectAnswerValueSet**

**Select dropdown for single selection**

Options are provided in the following ways:
- As contained valueSets in-questionnaire with a # reference in AnswerValueSet
- As a valueSet url in AnswerValueSet
- As an AnswerExpression

Does not support repeat answers
Fallbacks to an error alert if options cannot be rendered

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: unit, prompt, instructions

item.type == open-choice
**QItemOpenChoice**

item.type == open-choice
questionnaire-itemControl.code == radio
item.answerOption exists
**QItemOpenChoiceRadioAnswerOption**

**No AnswerValueSet variant,
Should probably build one**

**Radio buttons for single/multi selection with open label**

Options are provided in-questionnaire via AnswerOption
Supports repeat answers
Supports an "Other" option with user-specified free text
"Other" option supports debounce

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: instructions

item.type == open-choice
questionnaire-itemControl.code == checkbox
item.answerOption exists
**QItemOpenChoiceRadioAnswerOption**

**No AnswerValueSet variant,
Should probably build one**

**Checkboxes for single/multi selection with open label**

Options are provided in-questionnaire via AnswerOption
Supports repeat answers
Supports an "Other" option with user-specified free text
"Other" option supports debounce

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: instructions

item.type == choice
questionnaire-itemControl.code == autocomplete
**QItemOpenChoiceAutocomplete**

**Autocomplete for single selection - supports free text**

Options are provided by querying a terminology server (i.e. Ontoserver) with the valueSet **Sexpand** operation
Provides the top 10 results as options to choose from
Supports free-text as answer

Does not support repeat answers
Fallbacks to an error alert if options cannot be rendered

Supported properties: readOnly
Supported rendering extensions: choiceOrientation
Supported additional displayCategories: unit, prompt, instructions

item.type == open-choice
questionnaire-itemControl.code == select
item.answerValueSet exists
**QItemOpenChoiceRadioAnswerValueSet**

**Select dropdown for single selection with open label**

This component is not used in any of our questionnaires, therefore **UNTESTED**

**I dont think this component is implemented correctly at the moment**

item.type == open-choice
questionnaire-itemControl.code == select
item.answerValueSet doesn't exist
**QItemOpenChoiceRadioAnswerOption**

**Select dropdown for single selection with open label**

This component is not used in any of our questionnaires, therefore **UNTESTED**

**I dont think this component is implemented correctly at the moment**

*Open label - Component provides an "Other" option with free-text text field

**QItemRepeat**

- Repeat wrapper around a QItemSwitcher
- Allows for adding items and deleting items
- Contains states for repeat items

**QItemRepeatGroup**

- Repeat wrapper around a QItemGroup
- Allows for adding items and deleting groups
- Contains states for repeat groups

**QItemGroupTable**

- Is really a fancy QItemRepeatGroup structured in a table layout
- Each item represents a column
- Additional groups created represents a row of items