

FHIR Liquid Profile

This page documents how FHIR uses the [\[Liquid language\]](#) for templating.

The liquid templating engine is used for the following purposes:

- Building narratives for resources
- Building resources from external data sources (V2, CDA, etc)

Broadly, FHIR applications that use Liquid templating:

- Use a small subset of the liquid language (the syntax, and none of the functions)
- Replace liquid expressions with FHIRPath expressions

Contents

- [Evaluation Context](#)
- [Using Liquid Statements](#)
- [Statement](#)
- [Comments](#)
- [If](#)
- [Loop](#)
- [For](#)
 - [Else](#)
 - [Break](#)
 - [Continue](#)
 - [Cycle](#)
 - [Limit](#)
 - [Offset](#)
 - [Range](#)
 - [Reversed](#)
- [Include](#)
- [todo](#)

Evaluation Context

Whenever a Liquid template is executed, there's a single object that is the focus that the template is occurring in.

- When generating a resource, the resource itself is the focus.
- In other contexts, the focus object must be explicitly specified.

In addition to the focus object, other variables may be provided as specified (where?)

Using Liquid Statements

The following liquid statements are used, with the functionality as documented

Statement

A statement consists of 2 pairs of braces with a FHIRPath statement:

```
{{ expression }}
```

- Whitespace may be present before or after the expression statement, and is not significant.
- The expression is executed in the context of the focus object (but may reference other variables)

Comments

```
Anything you put between {% comment %} and {% endcomment %} tags  
is turned into a comment.
```

Allows you to leave un-rendered code inside a Liquid template. Any text within the opening and closing comment blocks will not be output, and any Liquid code within will not be executed.

If

The if control tag is

```
{% if expression %}  
{% else %}  
{% endif %}
```

- The expression is executed in the context of the focus object (but may reference other variables)
- Whitespace may be present in the tags, and is not significant.
- the else tag is optional
- {% elsif expression %} is also allowed

Loop

Note: Loop is deprecated in favor of the For control flow tag

The loop control tag is

```
{% loop var in expression %}  
  {{ var.expression }}  
{% endloop %}
```

- The loop will run once for each object returned by the expression
- The loop variable is available to expressions using the named parameter
- Whitespace may be present in the tags, and is not significant.

For

Note: For implementation is in progress as of 5/22/2020

The for control tag is

```
{% for var in expression %}  
  {{ var.expression }}  
{% endfor %}
```

- The for loop will run once for each object returned by the expression
- The for loop variable is available to expressions using the named parameter
- Whitespace may be present in the tags, and is not significant.

Else

The else tag specifies a fallback for a for loop which has no elements

```
{% for var in expression %}  
  {{ var.expression }}  
{% else %}  
  The collection is empty.  
{% endfor %}
```

Break

The break tag is used to break out of a loop

```
{% for var in expression %}  
  {{ var.expression }}  
  {% if expression %}  
    {% break %}  
  {% endif %}  
{% endfor %}
```

Continue

The continue tag is used to continue to the next iteration of a loop

```
{% for var in expression %}  
  {{ var.expression }}  
  {% if expression %}  
    {% continue %}  
  {% endif %}  
{% endfor %}
```

Cycle

The cycle tag is used to cycle through a list of items within each iteration of a for loop

```
{% for var in expression %}
  {{ var.expression }}
  {% cycle "odd", "even" %}
{% endfor %}
```

Limit

The limit parameter is used in a for loop to limit the number of iterations

```
{% for var in expression limit:2 %}
  {{ var.expression }}
{% endfor %}
```

Offset

The offset parameter is used in a for loop to skip a specified number of iterations

```
{% for var in expression offset:2 %}
  {{ var.expression }}
{% endfor %}
```

Range

Liquid defines a way to iterate a range of numbers (for i in (3..5)) but this is ambiguous with FHIRPath grammar and not supported.

Reversed

The reversed tag reversed the order of the elements in the loop

```
{% for var in expression reversed %}
  {{ var.expression }}
{% endfor %}
```

Include

The include control tag works like this:

```
{%include filename parameters %}
```

- the filename resolution is outside this spec
- Whitespace may be present in the tags, and is not significant.
- the parameters is a list of name=expression pairs, separated by whitespace
- the parameters are available in the included file using the variable "include"

e.g. the following include:

```
{%loop name in Patient.name%}{%include humannname.html name=name%}{%endloop%}
```

would lead to a statement like this in the included file:

```
{{include.name.family}}
```

todo

- What other liquid control things should be supported?
- date syntax wrangling