

Freemarker Reports Quick Reference Guide

Testing for nulls

By default, Freemarker will throw an exception if an object being printed to the report is null unless it is handled. This can be done with an `if` check, or using the null check.

A double question mark is used within an if statement to check that the value is not null, e.g.

```
<#if userName??>
    Hi ${userName}, How are you?
</#if>
```

This checks if the attribute of the object is not null:

```
<#if object.attribute??></#if>
```

This checks if object or attribute is not null (use parentheses):

```
<#if (object.attribute)??></#if>
```

Alternatively, an exclamation can be used for inline checking:

```
Hi ${userName!}, How are you?
```

This also can be used to provide a default value:

```
Hi ${userName!"John Doe"}, How are you?
```

If/Else Statements

Basic syntax is

```
<#if condition>
    ...
<#elseif condition2>
    ...
<#elseif condition3>
    ...
...
```

```
<#else>
    ...
</#if>
```

For example:

```
<#if x == 1>
    x is 1
<#else>
    x is not 1
</#if>
```

See [if, else, elseif - Apache FreeMarker Manual](#)

Loops/Iterating

Simplest form of a loop syntax is:

```
<#list sequence as item>
    Part repeated for each item
</#list>
```

Using this with a variable, this example looks like:

```
<#list users as user>
    <p>${user}</p>
</#list>
```

Print something when the list is empty:

```
<#list sequence as item>
    Part repeated for each item
<#else>
    Part executed when there are 0 items
</#list>
```

Do some processing when the list is not empty, before or after you start iterating, e.g. a header row in a table, or a summary row:

```
<#list sequence>
```

```
    Part executed once if we have more than 0 items
    <#items as item>
    Part repeated for each item
    </#items>
    Part executed once if we have more than 0 items
<#else>
    Part executed when there are 0 items
</#list>
```

See [list, else, items, sep, break, continue - Apache FreeMarker Manual](#)

Assigning variables within a report

Variables can be assigned with the `assign` directive, e.g.

```
<#assign total = 0>
```

These can then be manipulated, .e.g within a loop:

```
<#assign total += bean.amount>
```

And then just read back out later when they need to be printed

```
${total}
```

See [assign - Apache FreeMarker Manual](#)

Formatting values for display

Freemarker uses the built in Java formatter based on the value type. Sometimes this format is not appropriate, say you have an ID number which is a long, e.g. 123456. In Australia, the default numerical locale will print this as 123,456. To force this into a specific string format, we can specify:

```
${bean.longId?string["#####"]}
```

There are also many [built in formatters for dates](#), e.g.

```
${.now?string["dd-MMM-yyyy"]}
```

A format can be specified at the start of a template to be used for the rest of the report, for example

```
<#setting number_format="",##0.00">
```

Skyve also includes some custom Freemarker directives which can be used to use the formatter specified for the customer, e.g. date formatting, based on which dataset you are using.

BizQL or Bean dataset:

```
<@format bean=account binding="openDate" />
```

Note: do not put quotes around the bean, this is expecting an object to be passed through. This can be a compound binding.

SQL dataset:

```
<@sqlformat value=bean.dateCreated />
```

Including one template in another

A template can import another template with the `include` directive, e.g.

```
<#include "header.ftlh">
```

The path needs to be something Freemarker can find on the filesystem or in the database.

Macros

If you've mastered the basics, macros defined within a template can be useful for repeating rows within tables, etc.

See [macro, nested, return - Apache FreeMarker Manual](#)

The `import` directive can also be used to load a file containing re-usable macros but without outputting anything to the template like `include` does.

For example:

```
<#import "/libs/commons.ftl" as com>
<@com.copyright date="1999-2002"/>
```

See [import - Apache FreeMarker Manual](#)

Newlines

To preserve newlines in the HTML report for Freemarker, e.g. from a memo field in Skyve:

```
<#noautoesc>
    ${bean.comments!?replace('\n', '<br>')}
</#noautoesc>
```