Creating themes for LibreCCM with Freemarker

Starting with version 2.5 the LibreCCM platform support Freemarker as an alternative to XSL. Freemarker is a project of the Apache Foundation and a well known and mature template engine for Java. The support for Freemarker in version 2.5 is a backport from the upcoming version 7 of the LibreCCM platform.

Compared to XSL Freemarker is a lot easier to use, especially if you have worked with other template engines like Twig, Velocity etc before. In version 7 of the LibreCCM platform Freemarker will be become the primary template engine. XSL will still be supported, but we recommanded that you port your themes to Freemarker. Why Freemarker and not one of the other template engines? Freemarker is able to process XML in a similar way than XSL.

Freemarker also allows it to define user defined directivies and functions. To make it easier to create impressive themes we provide functions and macros for Freemarker we provide several functions and macros which hide the complexity of the XML data model created by CCM from the template author. It is recommanded not to access the XML data model directly. Instead the provided functions should be used. Otherwise your theme might brake when the XML structure changes.

General structure of a Freemarker theme

Freemarker themes have a different structure than the usual "old style" themes of LibreCCM.

ToDo

Predefined variables and functions

Several variables and functions are predefined and available without importing another file.

Variables

`contextPath`

The context path in which CCM is running.

`contextPrefix`

The context prefix.

dispatcherPrefix

Prefix for the CCM dispatcher (usually `/ccm`)

host

The current host.

model

The XML document created by LibreCCM.

negotiatedLanguage

The language negoiated between the user agent and LibreCCM.

requestScheme

The protocol (http or https).

selectedLanguage

The language selected by the user.

themePrefix

The prefix of the theme. Only available if a development theme is viewed.

Functions

getLocalizedText

`String getLocalizedText(String key)`

Returns the localized text from the resource bundle of the theme.

getContentItemTemplate

`String getContentItemTemplate(String objectType, view="DETAIL", style="")`

This is an internal function!

Returns the path for the template of a content item of a specific type.

_formatDateTime

An internal functions date time formatting. This functions should not be used directly.

Functions and Macros

Common functions

Language related

Import path: `<#import /utils.ftl as Utils>`

Sequence getAvailableLanguages

Sequence getAvailableLanguages()

Returns the available languages for the current document as sequence. These sequence can be used for creating links for selecting the language:

```
<ul class="language-selector">
    <#list Lang.getAvailableLanguages()?sort as lang>
        <li class="${(lang==negotiatedLanguage)?then('selected', '')}">${lang}</li>
    </#list>
</ul>
```