

Freie Markierung

Die Templating-Engine FreeMarker im Einsatz

Stefan Reinhold

stefan.reinhold@it-informatik.de

www.it-informatik.de

- Zusammenführen von Vorlagen und Daten zum Dokument

```
<h1>
  Hello ${name}!
</h1>
```

```
{
  "name" : "Joe"
}
```

```
<h1>
  Hello Joe!
</h1>
```

- Dadurch Trennung von Layout und Daten

- Reine Java-Bibliothek
- Entworfen im Hinblick auf Generierung von Webseiten
- Mächtige Template-Sprache
- Wird in vielen Softwareprojekten eingesetzt
- BSD-artige Lizenz („BSD-style license“)

- Geeignet zur Generierung beliebiger Textformate
- Leichtgewichtig (keine Abhängigkeiten)
- Übliche Kontrollstrukturen (if, switch, loop)
- Einfaches Definieren und Ändern von Variablen im Template
- Mächtige Ausdrücke
- Leichte Erstellung eigener Direktiven und Funktionen

- Einfacher und leichtgewichtiger als FreeMarker
- Weniger mächtige Template-Sprache
 - teilweise ThirdParty-Unterstützung notwendig
- Größere Community
 - Aber: für FreeMarker oft nicht notwendig

- Täglich aktualisierte Preisliste
- Artikeldaten als CSV-Datei (preise.csv)
- HTML-Gerüst (preise.html)
- Statische HTML-Seite als Ziel (preise.html)

- Frontend FMPP (FreeMarker-based text file PreProcessor)
 - BSD-artige Lizenz
 - Kommandozeilen-Schnittstelle
 - Ant-Task
 - Nutzbar als Java-Bibliothek
- Features
 - Konfiguration über Dateien möglich
 - Batch-Verarbeitung ganzer Verzeichnisbäume möglich
 - Daten-Lader für verschiedene Formate
 - Ausgabe konfigurierbar

- Daten-Lader für Formate
 - CSV
 - TDD (vermutlich: „Textual data definition“, ähnlich JSON)
 - Java- und Ant-Properties
 - Text
 - XML
- Angabe der Dateikodierung möglich
- Bereitstellung der Daten als Sequenzen und Hashes

- Text wird direkt ausgegeben
- Kommentare (`<!-- Kommentar -->`) werden entfernt
- Ausdrücke werden ausgewertet und deren Ergebnis ausgegeben
- Direktiven werden ausgewertet
 - Für Inhalt („Body“) der Direktiven gelten gleiche Regeln wie oben
- Unterdrückung unnötiger White-Space-Zeichen konfigurierbar

- Skalare Typen
 - String, Number, Boolean, Date
- Container
 - Hash, Sequence, Collection
- XML Tree
 - Node

- Generell: Groß-Klein-Schreibung beachten!
- Ausgabe von Variablen: `${variablenName}`
- Zugriff auf Hash-Werte: `hashset.key` **oder** `hashset[key]`
- Zugriff auf Sequenz-Werte: `seq[idx]`, `seq[start..end]`
- Zugriff auf Hierarchien: `parent.child.grandchild`

- Built-ins typabhängig
- Nutzung `var?built_in` bzw. `var?built_in(param)`
- Beispiele:
 - `"abc"?contains("b") → true`
 - `0.5?round → 1`
 - `["ice", "cream"?seq_contains("ice") → true`
 - `{key1 : value1, key2 : value2}?keys?sort
→ [key1, key2]`

- **Syntax:**
 - `<#directive arguments>...</#directive>`
 - `<#directive> oder <#directive />`
- **Einige Direktiven:**
 - Ablaufsteuerung (if, switch, list)
 - Variablendefinition (assign, global, local)
 - Ausgabeformatierung (compress, t, lt, rt, nt)
 - Definition und Nutzung von Makros, Funktionen
 - Templateverarbeitung (attempt, recover, stop)

■ Definition

- `<#macro myMacro param1 param2=defaultValue>`
 ...
 `<#nested>`
 ...
 `<#return>`
 `</#macro>`

■ Nutzung

- `<@myMacro param1=value1>...</@myMacro>`
- `<@myMacro />`

■ Definition

- `<#function myFunc arg1 arg2=defaultValue>`
 ...
 `<#return result>`
 `</#function>`

■ Nutzung

- `#{myFunc (arg1) }`

■ Keine Textausgabe

■ Rückgabewert erforderlich

- Import mit Namespace

- `<#import path/to/macrolib.ftl as m>`

- Nutzung mit Namespace

- `<@m.myMacro />`

- `#{m.myFunc () }`

- Statische Firmenwebsite einer fiktiven Unternehmung
- zweisprachig (englisch, deutsch)

- Einbindung von freemarker.ext.servlet.FreeMarkerServlet
- JSP-Taglibs nutzbar (Einbindung wie Makro)
 - `<#assign my=JspTaglibs["/my.tld"] />`
 - `<@my.myTag />`
 - Zahlenwerte sind `java.lang.BigDecimal`
- Servlet-Objekte verfügbar (Session, Request, Application)
- Referenzprojekt jforum (<http://jforum.net/>)

- Anwendungsfall
 - CMS generiert 4-stündlich ca. 3000 Webseiten
 - Webseiten enthalten dynamische Anteile
 - Google-Bot indiziert regelmäßig
- Lösung JSP
 - Sehr hoher Heap-Bedarf
 - Probleme bei Seiten mit langen Inhalten (Sitemap) mit 64k-Limit
- Lösung FreeMarker
 - 512 MB Heap ausreichend
 - Kein 64k-Limit

- Freemarkter (<http://freemarkter.org>)
- FMPP (<http://fmpp.sourceforge.net>)

Vielen Dank!

stefan.reinhold@it-informatik.de