

Frontend Technologie

Thymeleaf vs. AngularJS

PubliReport

Author: Guido Amrein / Urs Kehrl

Datum: 30. Oktober 2014

Inhalt

Technologie	2
i18n	2
Security	2
Validierung/Messages	3
Upload	3
Verschiedenes	3
Informationen	4
AngularJS	4
JHipster: Technology stack on the client side	4
JHipster: Technology stack on the server side	4
A complete Spring application	4
Ready to go into production	4

Technologie

Bereich	Thymeleaf	AngularJS
Rendering	Template serverseitig	Template clientseitig
Ajax Response	HTML - Fragment	JSON - String

i18n

Bereich	Thymeleaf	AngularJS
Labeling	Java Properties - File Bsp.: <code>#{LBL_name}</code>	JSON - File (kann aus dem Standard Java Properties-File generiert werden). Ext. Modul angular - translate Bsp.: <code><h1 translate="LBL_name"></h1></code>
Formatierungen (Datum, Zahlen, ...)	Spring Conversion Service Bsp.: <code>\${{val}}</code> Eigene Funktionen (ViewUtil)	Für Datum, Zahlen, Währung Modul vorhanden Bsp.: <code>{{ 1000 currency:"USD\$"/></code>
Sprachwahl	Serverseitig: Spring-Funktionalität (Page Refresh)	Clientseitig: Sprach-JSON-File dyn. nachgeladen Page Refresh

Security

Bereich	Thymeleaf	AngularJS
Security Provider	Spring - Security	Spring - Security
Login/Logout	Standard Spring - Security	Spring - Security mit AJAX-Erweiterungen
Remember Me	Ja	Ja
Taglib	Vorhanden <code><div sec:authorize="hasRole('ROLE_USER')"> ... content </div></code>	Über Direktiven <code><div ng-switch="authenticated"> <div ng-switch-when="true"> ... content </div> </div></code>

Validierung/Messages

Bereich	Thymeleaf	AngularJS
Technik	Serverseitig	Clientseitig (wo möglich)
Form Validation	Spring MVC, Bean Anotationen	Direktiven ng-minlength=5 ng-maxlength=100 required
Messages	<pre><p th:if="{#fields.hasErrors('date')}" th:errors="*{date}"> ... Incorrect date </p></pre>	<pre><p class="help-block" ng- show="form.email.\$error.email"> Your e-mail is invalid. </p></pre>

Upload

Bereich	Thymeleaf	AngularJS
Client	Standard File-Upload oder JQuery-Plugin	Verschiedene Module (basierend auf HTML5)
Server	Spring MVC (multipart resolver)	Spring MVC (multipart resolver)

Verschiedenes

Bereich	Thymeleaf	AngularJS
SEO*	Bei SPA* Sitemap erstellen	Bei SPA* Sitemap erstellen

SEO* (Search Engine Optimization)

SPA* (Single Page Application)

Allgemeines/Problematik zu SPA (Single Page Application) und SEO (Search Engine Optimization) (sollte seit Google beim Laden der Seite auch JavaScript ausgeführt entschärft sein):

<http://stackoverflow.com/questions/13499040/how-do-search-engines-deal-with-angularjs-applications>

Informationen

AngularJS

Gutes Beispiel-Projekt mit ganzem Tech Stack. <https://github.com/jhipster/jhipster-sample-app>

Diese Beispiel-App ist mit JHipster generiert: <http://jhipster.github.io/>

JHipster: Technology stack on the client side

- SPA Single Web Page Application
- Responsive Web Design
- [HTML5 Boilerplate](#)
- [Twitter Bootstrap](#)
- [AngularJS](#)
- Full internationalization support with [Angular Translate](#)
- Optional [Compass](#) / Sass support for CSS design
- Optional WebSocket support with the [Atmosphere framework](#)

And what if a single Web page application isn't enough for your needs?

Support for the [Thymeleaf](#) template engine, to generate Web pages on the server side

JHipster: Technology stack on the server side

A complete Spring application

- [Spring Boot](#) for easy application configuration
- [Maven](#) or [Gradle](#) configuration for building, testing and running the application
- [Development and production profiles](#) (both for Maven and Gradle)
- [Spring Security](#)
- [Spring MVC REST](#) + [Jackson](#)
- Optional WebSocket support with the [Atmosphere framework](#)
- [Spring Data JPA](#) + Bean Validation
- Database updates with [Liquibase](#)
- [MongoDB](#) support if you'd rather use NoSQL instead of a classical relational database

Ready to go into production

- Monitoring with [Metrics](#)
- Caching with [ehcache](#) (local cache) or [hazelcast](#) (distributed cache)
- Optional HTTP session clustering with [hazelcast](#)
- Optimized static resources (gzip filter, HTTP cache headers)
- Log management with [Logback](#), configurable at runtime
- Connection pooling with [HikariCP](#) for optimum performance
- Builds a standard WAR file or an executable JAR file