

Generierung einer frontendseitigen GWT Anwendung unter Verwendung von dem MVP Pattern und Dependency Injection

Claudia Schäfer | Marcus Fabarius | Stephanie Lehmann
CMS

10. Dezember 2013

INHALTSVERZEICHNIS

1. <i>Einleitung</i>	2
2. <i>Grundlagen</i>	3
2.1 Model Driven Architecture	3
2.1.1 Platform Independent Model	3
2.1.2 Platform Specific Model	3
2.2 GWT	4
2.2.1 MVP	4
2.2.2 UI-Binder	4
2.3 Dependency Injection mittels GIN	4
3. <i>Konzeption</i>	5
4. <i>UML Profil auf M2 Ebene</i>	6
5. <i>Aufbau und Struktur M1 Modell</i>	7
6. <i>Generator</i>	8
7. <i>Ergebnis</i>	9
8. <i>Fazit und Ausblick</i>	10

1. EINLEITUNG

2. GRUNDLAGEN

Text...

2.1 *Model Driven Architecture*

Model Driven Architecture (dt. Modellgetriebene Architektur), kurz MDA genannt, stellt einen bestimmten Ansatz zur Softwareentwicklung dar. Dieses Konzept ist 2001 von der Object Management Group (OMG) veröffentlicht worden und gilt heute als Standard. Hierbei werden Richtlinien zur Spezifikation in Form von Modellen vorgegeben. Aus diesen Modellen, die formal eindeutig sind, wird dann mithilfe von Generatoren automatisch der benötigte Code erzeugt. Ziel der MDA-Architektur ist es den gesamten Prozess der Softwareerstellung in Modellen darzustellen, so dass die Software zu einem hohen Anteil generativ, durch Transformationen von Modellen, erzeugt werden kann. Die dabei entstehenden Transformatoren können eine hohe Wiederverwendbarkeit und Wartbarkeit sicher stellen. Bei den Modellen handelt es sich im Speziellen um das Platform Independent Model und das Platform Specific Model, welche bei diesem Projekt auf dem Metamodell der UML basieren. Was dies genau bedeutet und wie die verschiedenen Modelle zu verstehen sind, wird in den folgenden Abschnitten erläutert.

2.1.1 *Platform Independent Model*

Das Platform Independent Model (PIM, dt. Plattformunabhängiges Modell) stellt ein Softwaresystem, das unabhängig von der technologischen Plattform ist, dar. In dem PIM sind alle Anforderungen erfasst, alles was es zu spezifizieren gibt im System ist definiert, jedoch komplett frei von der später folgenden Implementierung. Somit ist nicht nur eine bestimmte Implementierung des Systems möglich, sondern durchaus mehrere unterschiedliche.

2.1.2 *Platform Specific Model*

Kombiniert man die Funktionalitäten, die im Platform Independent Model definiert sind, mit den Designanforderungen der gewünschten Plattform, so erhält man das Platform Specific Model (PSM, dt. Plattformspezifisches Modell). Im Gegensatz zum PIM, welches nur die fachlichen Anforderungen definiert, werden beim PSM auch die technischen Aspekte eingebunden.

2.2 GWT

Das Google Web Toolkit, kurz GWT, ist ein open-source Projekt von Google. GWT ist ein Framework, welches genutzt werden kann um Webanwendungen mittels der Programmiersprache Java zu implementieren. Dabei übersetzt GWT den gesamten Source-Code in JavaScript Code und DOM-Elemente. Während der Übersetzung des Java Codes zu JavaScript Code werden darüber hinaus Optimierungen vorgenommen wie das Löschen von dead-Code. Dies führt potenziell dazu, dass komplexe Anwendungen im Browser schneller ausgeführt werden können. Darüber hinaus bietet GWT noch weitere Möglichkeiten, die dem Entwickler einer Webanwendung zu Gute kommen. Dazu zählen u. A. das Integrieren von JavaScript Code oder von JavaScript Bibliotheken innerhalb des Java Codes durch das JavaScript Native Interface, kurz JSNI und das sogenannte Code-Splitting, welches einem Entwickler ermöglicht sogenannte Split Points innerhalb des Codes zu setzen, welche dazu führen, dass bei der Ausführung der Anwendung bestimmte Inhalte ab dem Split Point später nachgeladen werden und dadurch die Startladezeit verringern. Google bietet mit zu den genannten Eigenschaften weitere positive Software Engineering Aspekte. Durch GIN (GWT INjection)

2.2.1 MVP

2.2.2 UI-Binder

2.3 Dependency Injection mittels GIN

3. KONZEPTION

4. UML PROFIL AUF M2 EBENE

5. AUFBAU UND STRUKTUR M1 MODELL

6. GENERATOR

7. ERGEBNIS

8. FAZIT UND AUSBLICK