

Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt  
Fakultät Informatik und Wirtschaftsinformatik

**Bachelorarbeit**

# **Design und Implementierung eines Generators für Android View Komponenten**

**vorgelegt an der Hochschule für angewandte Wissenschaften  
Würzburg-Schweinfurt in der Fakultät Informatik und Wirtschaftsinformatik zum  
Abschluss eines Studiums im Studiengang Informatik**

Marcel Groß

Eingereicht am: 31.03.2017

Erstprüfer: Prof. Dr. Peter Braun  
Zweitprüfer: Prof. Dr. Steffen Heinzl

## **Zusammenfassung**

TODO

## **Abstract**

TODO

# Danksagung

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Motivation . . . . .	3
1.2	Zielsetzung . . . . .	3
1.3	Aufbau der Arbeit . . . . .	4
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	REpresentational State Transfer (REST) . . . . .	5
2.1.1	Hypermedia as the Engine of Application State (HATEOAS) . . .	5
2.2	Android . . . . .	7
2.2.1	Custom Views . . . . .	7
2.3	Generatoren . . . . .	7
2.3.1	GEnerierung von Mobilen Applikationen basierend auf REST Ar- chitekturen (GeMARA) . . . . .	7
<b>3</b>	<b>Problemstellung</b>	<b>8</b>
<b>4</b>	<b>Lösung</b>	<b>9</b>
<b>5</b>	<b>Evaluierung</b>	<b>11</b>
<b>6</b>	<b>Zusammenfassung</b>	<b>13</b>
	<b>Verzeichnisse</b>	<b>14</b>
	<b>Literatur</b>	<b>15</b>
	<b>Eidesstattliche Erklärung</b>	<b>16</b>

# 1 Einführung

Das Smartphone ist heutzutage der stete Begleiter eines Menschen. „Zwei Drittel der Bevölkerung und nahezu jeder 14- bis 29-Jährige geht darüber ins Netz.“ [usage] Auch die Prognose zeigt, dass der Absatzmarkt immer weiter steigen wird (Abbildung 1.1).

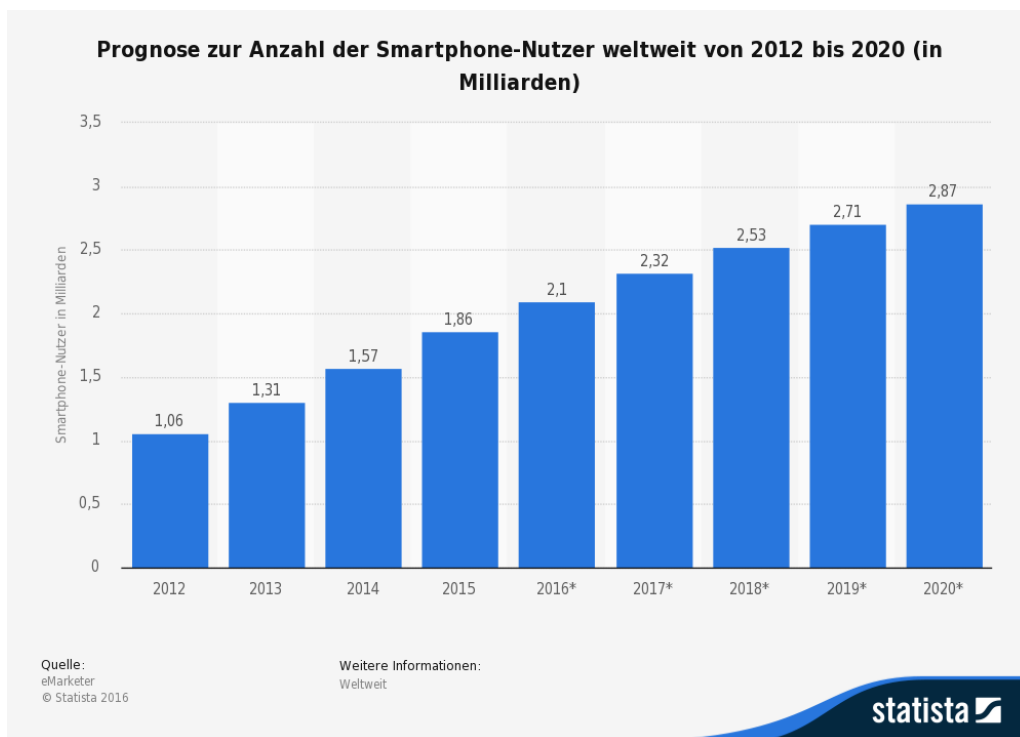


Abbildung 1.1: Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2020 (in Milliarden) [prognose].

Umso wichtiger ist es, dass die Softwareentwicklung diesen Trend ernst nimmt. Der ehemalige Google-Chef Eric Schmidt sagte bereits 2010: „Googles Devise heißt jetzt ‚Mobile first‘“. Diese Devise wird von vielen Unternehmen verfolgt, das ist der Grund, weswegen in den einzelnen Stores heutzutage so viele Apps angeboten werden. Bei Android im Playstore sind es im Oktober 2016 ca. 2,4 Millionen Apps [play\_store], bei Apple im App Store sind es ca. 2 Millionen Apps (Stand Juni 2016) [app\_store]. Neben Googles

Android und Apples iOS gibt es noch andere Betriebssysteme, beispielsweise Microsofts Windows Phone oder Blackberrys Blackberrys OS. Jedoch bestimmen die beiden erstgenannten Systeme den Markt (Abbildung 1.2).

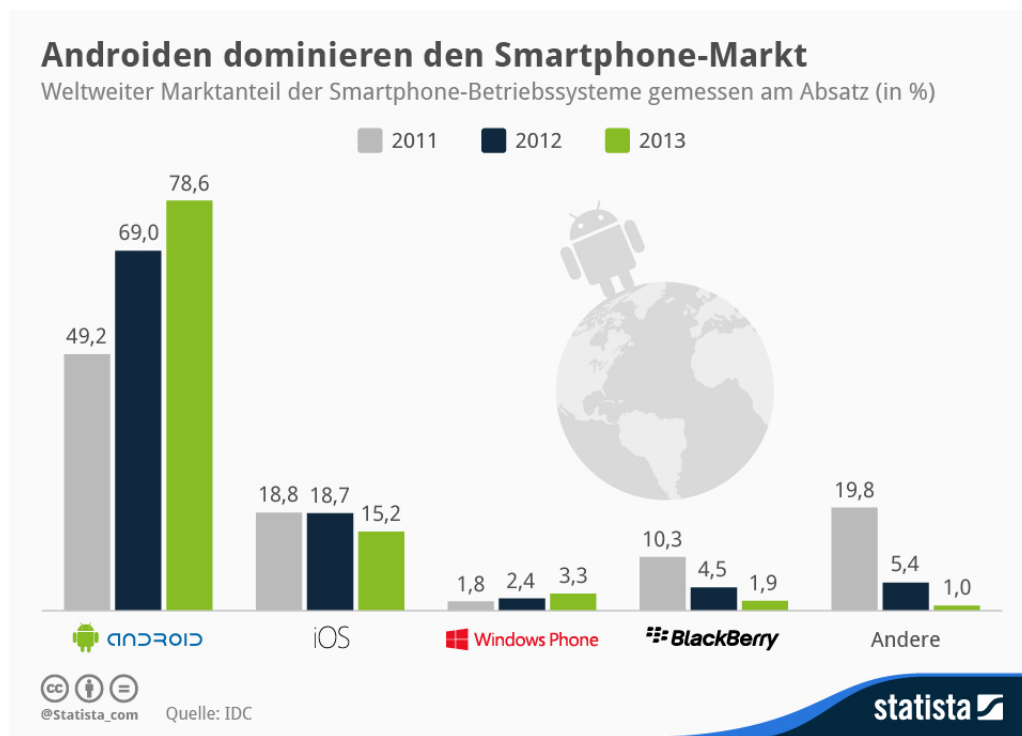


Abbildung 1.2: Der weltweite Marktanteil von Smartphone-Betriebssysteme. [os]

Jede dieser Applikationen wurden einzeln für sich entwickelt und implementiert. Bei jedem Update zum Beispiel des Systems, müssen alle Anwendungen gewartet und überarbeitet werden, um die volle Funktionalität zu gewährleisten.

Würden einige Applikationen jedoch genauer analysiert werden, wäre das Ergebnis, dass in jeder dieser Anwendungen Codepassagen vorhanden sind, welche einen ähnlichen beziehungsweise den selben Zweck erfüllen. Werden diese Stellen im Programmcode abstrahiert, gibt es die Möglichkeit diese generieren zu lassen. Um Code generieren zu lassen, benötigt man so genannte Code-Generatoren.

Im Bereich der Backend-Entwicklung gibt es bereits verschiedene Projekte die sich damit befassen. Ein Beispiel wäre der „CRUD Admin Generator“ [generators]. Die Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt entwickelt unter der Leitung von Prof. Dr. Peter Braun auch einen Code-Generator unter dem Namen: GeMARA. Mit Hilfe solcher Generatoren für den Bereich von Mobilen Applikationen, könnte der Entwicklungs- und Wartungsaufwand reduziert werden.

Führt ein Systemupdate dazu, dass die Implementierung von verschiedenen Anforderungen nicht weiter funktionsfähig ist, muss dies nur einmalig an der entsprechenden Stelle im Code-Generator geändert werden und nicht in jeder Applikation einzeln.

### 1.1 Motivation

Im Rahmen des Projektes GeMARA gab es bereits Arbeiten, welche sich mit dem Thema der Generierung von Android Activities beschäftigt. Die dabei entstandenen Lösungen, resultierten darin, dass das generieren von Activities zu Problemen führt. Deshalb beschäftigt sich diese Ausarbeitung damit, nicht eine komplette Activity zu erzeugen, sondern sogenannte Komponenten.

Eine Komponente, ist im wesentlichen eine kleine Anwendung für sich, welche nur eine einzige Aufgabe erfüllt. Dies könnte zum Beispiel das Anzeigen eines Dozenten in einer Campus-Applikation sein.

Aus den erzeugten Komponenten, kann eine Art „Bausatz“ entstehen. Mit dessen Hilfe der Entwickler seine Applikation zusammen bauen kann. Dabei wird ihm freie Wahl gelassen, wie der Aufbau seiner Anwendung aussieht, er bedient sich nur an gegebener Stelle an den Komponenten. Dadurch reduziert sich der Entwicklungsaufwand für ihn.

Bewegen wir uns in der Domäne einer Hochschule, so kann eine Bibliothek mit den erzeugten Komponenten allen Studierenden zur Verfügung gestellt werden. Dadurch wäre jeder dieser Studierenden in der Lage eine persönliche Campus-Applikation zu entwickeln. Durch die einzelnen Komponenten kann dann sichergestellt werden, dass grundsätzliche Funktionalität bereits gewährleistet ist.

### 1.2 Zielsetzung

Ziel dieser Ausarbeitung liegt darin, dass der Leser ein grundsätzliches Verständnis für die Entwicklung von Android-Applikationen beziehungsweise Android-Bibliotheken vermittelt bekommt. Weiterhin soll das Wissen über Datenkommunikation mittels REST vertieft werden. Hierbei wird ein Schwerpunkt auf das Hypermedia-Prinzip gelegt.

Neben diesen spezifischen Anforderungen, soll ein Verständnis für der Implementierung von Generatoren entstehen. Dafür muss der Entwickler entscheiden können, was von der Implementierung als statischer Code angesehen werden kann und welcher generisch ist.

Dieses Verständnis ist wichtig, um die Komplexität der Generatoren zu reduzieren. Da die statischen Anteile jedesmal identisch sind.

Auch soll auf die Frage eingegangen werden, ob man das User Interface (UI), welches ebenfalls generiert wird, auch generisch gestalten kann. Das bedeutet, dass nicht nur die Informationen, welche angezeigt werden sollen beschreibt. Sondern auch wie diese angezeigt werden sollen.

Wenn es möglich ist dass dasUI als Teil der domänenspezifische Sprache (DSL) beschrieben werden kann, so hat der Nutzer des entsprechenden Generators die Freiheit, selbst zu entscheiden ob zum Beispiel bei seiner Campus-App, bei der Liste aller Dozenten das Profilbild links oder rechts angezeigt werden soll.

### 1.3 Aufbau der Arbeit

Aufbau



## 2 Grundlagen

### 2.1 REpresentational State Transfer (REST)

In dem generierten Projekt, sollen alle benötigten Inforamtionen mittels REST von dem zugehörigen, generierten Backend geladen werden.

REST [**rest\_\_fielding**] ist ein Programmierparadigma, welches sich auf folgende Prinzipien stützt:

1. Client-Server
2. Zustandslose Kommunikation
3. Caching
4. Uniform Interface
5. Layered System
6. Code-on-Demand (optional)

#### 2.1.1 Hypermedia as the Engine of Application State (HATEOAS)

HATEOAS fällt unter das Prinzip Uniform Interface. Es beschreibt, wie mit Hilfe eines endlichen Automaten eine REST-Architektur entworfen werden kann.

Der Architekt einer REST-konformen Application Programming Interface (API) überlegt sich im vorraus, wie der Applikation-Fluss in der späteren Anwendung aussehen soll. Dafür definiert er verschiedene States und welche Transitionen zum nächsten State führen.

Als ein State kann beispielsweise das Anzeigen aller Lecturer in einer Campus-Applikation

angesehen werden. Die Transition hingegen ist zum Beispiel ein Link im Link-Header der Antwort, oder ein Attribut, der empfangen Ressource.

Wird das API mit Hilfe eines endlichen Automaten entwickelt, kann diese dem Client-Entwickler als Anleitung zum erstellen seines Clients dienen. Er benötigt lediglich einen Uniform Resource Locator (URL), welche auf den initialen State des endlichen Automaten führt. Dieser liefert dann alle, zu diesem Zeitpunkt möglichen, Transitionen zurück. Mit Hilfe dieser Transaktionen, kann sich der Entwickler dann zum nächsten State bewegen. Auch dieser State liefert alle möglichen weiteren Transaktionen zurück. Wenn der Entwickler sich so durch die States bewegt, bekommt er die benötigten Informationen zum Aufbau und Ablauf der Applikation.

Die Abbildung ?? zeigt einen solchen Automaten. Der Einstiegspunkt ist der State „Dispatcher“ dieser liefert die Transition zum State „Collection“ zurück. Dieser State, verfügt über alle Informationen die benötigt werden um eine Collection der betroffenen Ressource anzuzeigen, weiterhin verfügt er auch das Wissen, über die beiden nächsten Transitionen zu den States „Create“ und „Single“. Wie der Name des States annehmen lässt, wird der State „Create“ benötigt um eine neue Ressource anzulegen. Von diesem State aus kann die Anwendung nur zurück zum State „Collection“. Der State „Single“ enthält alle benötigten Daten um eine einzelne Ressource anzuzeigen. Von hier kann die Anwendung zum State „Update“ oder „Delete“ wechseln. Der State „Update“ ermöglicht es die Ressource zu bearbeiten. Von hier kann der Nutzer der Anwendung nur zum State „Single“ zurückkehren. Der State „Delete“ löscht die aktuelle Ressource und liefert die Transition zum State „Collection“ zurück. Dieses Beispiel verdeutlicht noch einmal bildlich, das der Entwickler nur den Einstiegspunkt „Dispatcher“ kennen muss. Die Anwendung liefert selbst alle benötigten Informationen um die Daten für die Anwendung nachzuladen.

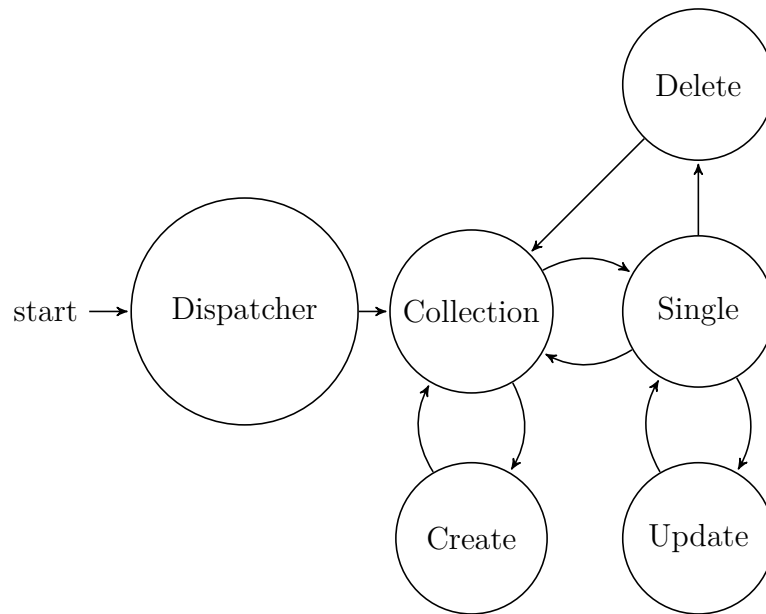


Abbildung 2.1: Aufbau eines REST-API mit Hilfe eines endlichen Automaten.

## 2.2 Android

### 2.2.1 Custom Views

Noch mehr Text

## 2.3 Generatoren

### 2.3.1 GEneration von Mobilen Applikationen basierend auf REST Architekturen (GeMARA)

## **3 Problemstellung**

## 4 Lösung

Listing 4.1: Beispiel für einen Quelltext

```
1  
2 public void foo() {  
3     // Kommentar  
4 }
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut vehicula felis lectus, nec aliquet arcu aliquam vitae. Quisque laoreet consequat ante, eget pretium quam hendrerit at. Pellentesque nec purus eget erat mattis varius. Nullam ut vulputate velit. Suspendisse in dui in eros iaculis tempus. Phasellus vel est arcu. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Integer elementum, nulla eu faucibus dignissim, orci justo imperdiet lorem, luctus consectetur orci orci a nunc.

Praesent at nunc nec tortor viverra viverra. Morbi in feugiat lectus. Vestibulum iaculis ipsum at eros viverra volutpat in id ipsum. Donec condimentum, ligula viverra pharetra tincidunt, nunc dui malesuada nisi, vitae mollis lacus massa quis velit. Integer feugiat ipsum a volutpat scelerisque. Nulla facilisis augue nunc. Curabitur eget consectetur nulla. Integer accumsan sem non nisi tristique dictum.

Sed lacinia eu dolor sed congue. Ut dui orci, venenatis id interdum rhoncus, mattis elementum massa. Proin venenatis elementum purus ut rutrum. Phasellus sit amet enim porta, commodo mauris a, bibendum tortor. Nulla ut lobortis justo. Aenean auctor mi nec velit fermentum, quis ultricies odio viverra. Maecenas ultrices urna vel erat ornare, quis suscipit odio molestie. Donec vel dapibus orci, vel tincidunt orci.

Etiam vitae eros erat. Praesent nec accumsan turpis, et mollis eros. Praesent lacinia nulla at neque porta aliquam. Quisque elementum neque ac porta suscipit. Nulla volutpat luctus venenatis. Aliquam imperdiet suscipit pretium. Nunc feugiat lacinia aliquet. Mauris ut sapien nec risus porttitor bibendum. Aenean feugiat bibendum lectus, id mattis elit adipiscing at. Pellentesque interdum felis non risus iaculis euismod fermentum nec urna. Nullam lacinia suscipit erat ac ullamcorper. Sed vitae nulla posuere, posuere sem id, ultricies urna. Maecenas eros lorem, tempus non nulla vitae, ullamcorper egestas nibh. Vestibulum facilisis ante vel purus accumsan mattis. Donec molestie tempor eros, a gravida odio congue posuere.

Sed in tempus elit, sit amet suscipit quam. Ut suscipit dictum molestie. Etiam quis porta mauris. Cras dapibus sapien eget sem porta, ut congue sapien accumsan. Maecenas hendrerit lobortis mauris ut hendrerit. Suspendisse at aliquet est. Quisque eros est, scelerisque ac orci quis, placerat suscipit lorem. Phasellus rutrum enim non odio ullamcorper, sit amet auctor nulla fringilla. Nunc eleifend vulputate dui, a sollicitudin tellus venenatis non. Cras condimentum lorem at ultricies vestibulum. Vestibulum interdum lobortis commodo. Nullam rhoncus interdum massa, ut varius nisi scelerisque id. Nunc interdum quam in enim bibendum vulputate.

## 5 Evaluierung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut vehicula felis lectus, nec aliquet arcu aliquam vitae. Quisque laoreet consequat ante, eget pretium quam hendrerit at. Pellentesque nec purus eget erat mattis varius. Nullam ut vulputate velit. Suspendisse in dui in eros iaculis tempus. Phasellus vel est arcu. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Integer elementum, nulla eu faucibus dignissim, orci justo imperdiet lorem, luctus consectetur orci orci a nunc.

Praesent at nunc nec tortor viverra viverra. Morbi in feugiat lectus. Vestibulum iaculis ipsum at eros viverra volutpat in id ipsum. Donec condimentum, ligula viverra pharetra tincidunt, nunc dui malesuada nisi, vitae mollis lacus massa quis velit. Integer feugiat ipsum a volutpat scelerisque. Nulla facilisis augue nunc. Curabitur eget consectetur nulla. Integer accumsan sem non nisi tristique dictum.

Sed lacinia eu dolor sed congue. Ut dui orci, venenatis id interdum rhoncus, mattis elementum massa. Proin venenatis elementum purus ut rutrum. Phasellus sit amet enim porta, commodo mauris a, bibendum tortor. Nulla ut lobortis justo. Aenean auctor mi nec velit fermentum, quis ultricies odio viverra. Maecenas ultrices urna vel erat ornare, quis suscipit odio molestie. Donec vel dapibus orci, vel tincidunt orci.

Etiam vitae eros erat. Praesent nec accumsan turpis, et mollis eros. Praesent lacinia nulla at neque porta aliquam. Quisque elementum neque ac porta suscipit. Nulla volutpat luctus venenatis. Aliquam imperdiet suscipit pretium. Nunc feugiat lacinia aliquet. Mauris ut sapien nec risus porttitor bibendum. Aenean feugiat bibendum lectus, id mattis elit adipiscing at. Pellentesque interdum felis non risus iaculis euismod fermentum nec urna. Nullam lacinia suscipit erat ac ullamcorper. Sed vitae nulla posuere, posuere sem id, ultricies urna. Maecenas eros lorem, tempus non nulla vitae, ullamcorper egestas nibh. Vestibulum facilisis ante vel purus accumsan mattis. Donec molestie tempor eros, a gravida odio congue posuere.

Sed in tempus elit, sit amet suscipit quam. Ut suscipit dictum molestie. Etiam quis porta mauris. Cras dapibus sapien eget sem porta, ut congue sapien accumsan. Maecenas hendrerit lobortis mauris ut hendrerit. Suspendisse at aliquet est. Quisque eros est, scelerisque ac orci quis, placerat suscipit lorem. Phasellus rutrum enim non odio ullamcorper, sit amet auctor nulla fringilla. Nunc eleifend vulputate dui, a sollicitudin tellus venenatis non. Cras condimentum lorem at ultricies vestibulum. Vestibulum interdum

## *5 Evaluierung*

lobortis commodo. Nullam rhoncus interdum massa, ut varius nisi scelerisque id. Nunc interdum quam in enim bibendum vulputate.



## **6 Zusammenfassung**

# Abbildungsverzeichnis

1.1	Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2020 (in Milliarden) [ <b>prognose</b> ]. . . . .	1
1.2	Der weltweite Marktanteil von Smartphone-Betriebssysteme. [ <b>os</b> ] . . . . .	2
2.1	Aufbau eines REpresentational State Transfer-Application Programming Interface mit Hilfe eines endlichen Automaten. . . . .	7

# Tabellenverzeichnis

# Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorgelegte Bachelorarbeit selbstständig verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt habe. Alle benutzten Quellen und Hilfsmittel sind angegeben, wörtliche und sinngemäße Zitate wurden als solche gekennzeichnet.

---

Marcel Groß, am 1. Dezember 2016