



Hochschule für
Technik und Wirtschaft
Dresden
University of Applied Sciences

Fakultät Informatik/Mathematik

Diplomarbeit

im Studiengang Allgemeine Informatik

Thema:

**Webentwicklung mittels Kotlin am Beispiel eines
RESTful-Schachservers**

Eingereicht von: Felix Dimmel

Eingereicht am: 28. März 2018

Betreuer: Prof. Dr.-Ing. Jörg Vogt

2. Gutachter: Prof. Dr.-Ing. Arnold Beck

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
3	Vergleich zwischen Kotlin und dem Google Web Toolkit	5
4	Konzeption des Servers	7
4.1	Anforderungen	7
4.1.1	Ressource: Player (Spieler)	7
4.1.2	Ressource: Match (Partie)	8
4.1.3	Ressource: Draw (Zug)	8
4.2	Verwendete Bibliotheken	8
4.2.1	Spring	8
4.2.2	SQLite	8
4.2.3	ORMLite	8
4.3	Ressourcenzugriffe mithilfe von Spring Controllern	8
4.3.1	Player Controller	8
4.3.2	Match Controller	8
4.3.3	Draw Controller	8
4.3.4	Game Controller	8
4.3.5	Error Controller	8
5	Konzeption des Clients	9
6	Implementation des Servers	11
7	Implementation des Clients	13
8	Fazit	15
9	Ausblick	17
	Literatur	19

Anhang	31
A First chapter of appendix	31
A.1 Parameters	31

KAPITEL 1

Einleitung

Laut einer Onlinestudie der ARD/ZDF-Medienkommission von 2017 [[ARD17a](#); [ARD17b](#)] ist die Nutzung des Internets im letzten Jahr um 6% in Deutschland gestiegen. Dieser große Bedeutungszuwachs allein in Deutschland zeigt auch die immer mehr zunehmende Nachfrage nach Webinhalten. Was wiederum auch ein Indiz für die große Beliebtheit der Skriptsprache JavaScript sein könnte, denn diese ist laut der letzten jährlichen Onlineumfrage der Stack Exchange Inc. [[Inc17](#)] auf Platz 1 der beliebtesten Programmiersprachen. Dennoch kann es mitunter schwierig werden große Applikationen mit JavaScript zu entwickeln. An dieser Stelle kommt die noch sehr junge Programmiersprache Kotlin ins Spiel.

Kotlin ist eine statisch typisierte objektorientierte Programmiersprache, welche in Java-Bytecode, in JavaScript und in Native Code kompiliert werden kann. Desweiteren ist Kotlin neben Java seit Mai 2017 offizielle Programmiersprache für Android [[Sha17](#)]. Dadurch ist Kotlin sehr vielseitig und lässt sich für viele Anwendungsfälle einsetzen.

Da aber Kotlin wie schon oben erwähnt noch sehr jung ist, soll diese Arbeit dazu genutzt werden zu erforschen wie sich diese Programmiersprache für Server- bzw. Client-Anwendungen eignet bzw. welche Probleme während der Umsetzung aufgetreten sind.

TODO:

- Kapitelverlauf erläutern
- Voraussetzungen

KAPITEL 2

Grundlagen

KAPITEL 3

Vergleich zwischen Kotlin und dem Google Web Toolkit

KAPITEL 4

Konzeption des Servers

Inhalt dieses Kapitels soll die Planung sein, welche für die Umsetzung des RESTful Schachservers benötigt wird. Dabei dient der erste Abschnitt für eine Erläuterung der Anforderungen, welche der Server mitbringen bzw. erfüllen soll. Im zweiten Abschnitt werden die verwendeten Bibliotheken in den Punkten Zweck, Einrichtung und Verwendung näher erläutert. Der letzte Abschnitt dieses Kapitels befasst sich anschließend damit, wie der Zugriff auf einzelne Ressourcen des REST-Server erfolgen soll. Dabei werden alle möglichen Request-Methoden für die jeweiligen Ressourcen näher beleuchtet.

4.1 Anforderungen

Die Grundanforderungen an den RESTful Schachserver sollen in erster Linie die Bereitstellung aller benötigten Ressourcen sein. Dabei sollen Elemente erstellt, ggf. bearbeitet und gelöscht werden können. Zusätzlich soll die Möglichkeit bestehen, einzelne oder alle gespeicherten Elemente einer Ressource abzufragen. Des weiteren sollen alle Ressourcenelemente in einer SQLite Datenbank gespeichert werden, welche für jedes Element eindeutige ID's generiert.

Um ein Schachspiel abzubilden bedarf es dabei der Ressourcen Player (Spieler), Match (Partie) und Draw (Zug), welche in den nachfolgenden Unterabschnitten 4.1.1 bis 4.1.3 näher betrachtet werden.

Als abschließende Anforderung ist noch die Fehlerresistenz zu erwähnen. Denn die im Rahmen dieser Arbeit entstandene Praktikumsaufgabe

Verweis auf Praktikumsaufgabe im Anhang

soll durch zukünftige Studenten absolviert werden, wobei Sie den Server als Grundlage gestellt bekommen sollen.

4.1.1 Ressource: Player (Spieler)

Neben der ID die wie schon in Abschnitt 4.1 erwähnt durch die SQLite Datenbank generiert werden soll, besitzt der Player noch die Information über den Name und das Passwort des Players.

Eine nachträgliche Änderung des Namens soll dabei nicht gestattet sein. Das Passwort hingegen soll zu jeder Zeit aktualisierbar sein.

4.1.2 Ressource: Match (Partie)

Auch bei der Ressource Match wird die ID durch SQLite generiert. Des weiteren besitzt ein Match Informationen über die beiden Spieler(Weiß/Schwarz) und deren Figurenstellung. Ein Match weiß außerdem welche Farbe am Zug ist, ob einem Spieler eine Rochade zur Verfügung steht, ob ein Feld existiert für ein Schlag en passant und wie viele Halbzüge getätigt wurde seit dem zuletzt ein Bauer gezogen oder eine Figur geschlagen wurde. Zusätzlich kann über ein Match ermittelt werden ob ein Spieler im Schach steht oder das Spiel schon bis zum Schachmatt gespielt wurde. Zuletzt hält das Match noch den aktuellen Stand in der Forsyth-Edwards-Notation (FEN).

4.1.3 Ressource: Draw (Zug)

Die Ressource Draw speichert zusätzlich zur ID die Farbe des Spielers, die Art der Spielfigur, Start- und Endfeld, ob eine Figur geschlagen wurde und wenn ja ob durch en passant und ob seitens der Dame oder des König rochiert wurde. Zusätzlich hält der Draw seine Daten in der Standard Algebraic Notation (SAN).

4.2 Verwendete Bibliotheken

4.2.1 Spring

4.2.2 SQLite

4.2.3 ORMLite

4.3 Ressourcenzugriffe mithilfe von Spring Controllern

4.3.1 Player Controller

4.3.2 Match Controller

4.3.3 Draw Controller

4.3.4 Game Controller

4.3.5 Error Controller

KAPITEL 5

Konzeption des Clients

KAPITEL 6

Implementation des Servers

KAPITEL 7

Implementation des Clients

KAPITEL 8

Fazit

KAPITEL 9

Ausblick

Literatur

- [ARD17a] ARD/ZDF-MEDIENKOMMISSION: „ARD/ZDF-Onlinestudie 2017: Neun von zehn Deutschen sind online. Bewegtbild insgesamt stagniert, während Streamingdienste zunehmen - im Vergleich zu klassischem Fernsehen jedoch eine geringe Rolle spielen.“ *Media Perspektiven* (Sep. 2017), Bd. URL: <http://www.ard-zdf-onlinestudie.de/ardzdf-onlinestudie-2017/> (besucht am 22. 03. 2018) (siehe S. 1).
- [ARD17b] ARD/ZDF-MEDIENKOMMISSION: „Kern-Ergebnisse der ARD/ZDF-Onlinestudie 2017“. *Media Perspektiven* (Sep. 2017), Bd. URL: http://www.ard-zdf-onlinestudie.de/files/2017/Artikel/Kern-Ergebnisse_ARDZDF-Onlinestudie_2017.pdf (besucht am 26. 03. 2018) (siehe S. 1).
- [Inc17] INC., STACK EXCHANGE: „Developer Survey 2017“. (2017), Bd. URL: <https://insights.stackoverflow.com/survey/2017#technology> (besucht am 26. 03. 2018) (siehe S. 1).
- [Sha17] SHAFIROV, MAXIM: „Kotlin on Android. Now official“. (Mai 2017), Bd. URL: <https://blog.jetbrains.com/kotlin/2017/05/kotlin-on-android-now-official/> (besucht am 26. 03. 2018) (siehe S. 1).

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Acknowledgments

I thank ?? and ?? for giving me the opportunity to write this bachelor/master/phd thesis at ??, and for their professional advise.

I thank in particular the ?? team who readily/willingly provided information at any time and ??.

I would also like to than all people who supported me in writing this thesis.

Erklärung der Selbstständigkeit

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie die Zitate deutlich kenntlich gemacht zu haben.

Dresden, den 28. März 2018

Felix Dimmel

A First chapter of appendix

A.1 Parameters

