



HTWK Leipzig

Fakultät für Informatik, Mathematik & Naturwissenschaften

Bachelor-Thesis

Generierung und Design einer Client-Bibliothek für einen RESTful Web Service am Beispiel der Spreadshirt-API

Author:

Andreas Linz

10INB-T

admin@klingt.net

Leipzig, 18. Juni 2013

Gutachter:

Dr. rer. nat. Johannes Waldmann

HTWK Leipzig – Fakultät für Informatik, Mathematik & Naturwissenschaften

waldmann@imn.htwk-leipzig.de

HTWK Leipzig, F-IMN, Postfach 301166, 04251 Leipzig

Jens Hadlich

Spreadshirt HQ, Gießerstraße 27, 04229 Leipzig

jns@spreadshirt.net

Andreas Linz
Nibelungenring 52
04279 Leipzig
admin@klingt.net
www.klingt.net

*Generierung und Design einer Client-Bibliothek für einen
RESTful Web Service am Beispiel der Spreadshirt-API*
Bachelor Thesis, HTWK-Leipzig, 18. Juni 2013

made with X_YT_EX, L^AT_EX and B_IB_TE_X.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich diese Bachelor-Thesis selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht.

Diese Arbeit ist bislang keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht worden.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

.....

Andreas Linz

Leipzig, 18. Juni 2013

Danksagungen

Mein Dank gilt Dr. rer. nat. Johannes Waldmann für die Betreuung der Arbeit, sowie Spreadshirt und insbesondere Jens Hadlich für die Unterstützung.

Ohne das Korrekturlesen von Elisa Jentsch hätte ich so manchem Fehler nicht entdeckt, dafür auch noch einmal vielen Dank.

Abstract

Schlüsselwörter

Codegenerierung, RESTful Web Service, Modellierung, Client-Bibliothek, Spreadshirt-API, Polyglot

Diese Seite wurde mit Absicht leer gelassen.

Inhaltsverzeichnis

1 Einführung	1
2 Grundlagen	2
2.1 RESTful Web Service	2
2.2 WADL	2
2.3 XSD	2
3 Implementierung	3
4 Zusammenfassung	4
4.1 Fazit	4
4.2 Ausblick	4
Glossar	A
Abbildungsverzeichnis	B
Tabellenverzeichnis	C
Listings	D
Literaturverzeichnis	E

Diese Seite wurde mit Absicht leer gelassen.

1 Einführung

„Essentially, all models are wrong, but some are useful.“

GEORGE E. P. BOX, NORMAN R. DRAPER

Empirical Model-Building and Response Surfaces. p. 424

Ziel dieser Bachelor-Thesis ist die Erzeugung und das Design einer Client-Bibliothek aus der abstrakten Beschreibung der Spreadshirt **REST API** durch einen Codegenerator.

2 Grundlagen

2.1 RESTful Web Service

2.2 WADL

2.3 XSD

3 Implementierung

4 Zusammenfassung

4.1 Fazit

4.2 Ausblick

Glossar

API

Application Programming Interface (deutsch: „Schnittstelle zur Anwendungsprogrammierung“) spezifiziert wie Softwarekomponenten über diese Schnittstelle miteinander interagieren können . [1](#), [A](#)

JSON

JavaScript Object Notation ist ein Mensch- und Maschinenlesbares Format zu Codierung und Austausch von Daten. Bietet im Gegensatz zu XML keine Erweiterbarkeit und Unterstützung für Namesräume, ist aber kompakter und einfacher zu parsen. [A](#), *siehe XML*

Metaprogramming

beschreibt das erstellen von Programmen welche sich selbst, oder andere Programme, modifizieren oder die einen Teil des Kompilierungsschrittes übernehmen (bspw. der C-Präprozessor) . [A](#)

Polyglot

mehrsprachig . [A](#)

REST

Representational State Transfer (deutsch: „Gegenständlicher Zustands-transfer“) ist ein Softwarearchitekturstil für Webanwendungen, welcher von Roy Fielding in seiner [Dissertation](#) ¹ beschrieben wurde. Die Daten liegen dabei in eindeutig adressierbaren *resources* vor. Die Interaktion basiert auf dem Austausch von *representations* – also ein Dokument was den aktuellen oder gewünschten Zustand einer resource beschreibt. Beispiel-URL für das Item 84 aus dem Warenkorb 42:

<http://api.spreadshirt.net/api/v1/baskets/84/item/42> . [1](#), [A](#)

RESTful

Als *RESTful* bezeichnet man einen Webservice der den Prinzipien von REST entspricht. [A](#), *siehe REST*

¹ http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf

WADL

Web Application Description Language ist eine maschinenlesbare Beschreibung einer HTTP-basierten Webanwendung. [A](#), *siehe* [XML](#)

XML

Extensible Markup Language (deutsch: „erweiterbare Auszeichnungssprache“) ist ein Mensch- und Maschinenlesbares Format für Codierung und Austausch von Daten, [spezifiziert vom W3C](#) ² . [A](#)

XSD

XML Schema Description enthält Regeln für den Aufbau und zum Validieren einer XML-Datei. [A](#), *siehe* [XML](#)

² <http://www.w3.org/TR/REC-xml>

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Literaturverzeichnis

- Czarnecki, K. und U. Eisenecker (2000). *Generative programming: methods, tools, and applications*. Addison Wesley. ISBN: 9780201309775. URL: <http://books.google.de/books?id=cCZXYQ6Pau4C>.
- Fowler, M. (2010). *Domain-Specific Languages*. Addison-Wesley Signature Series (Fowler). Pearson Education. ISBN: 9780131392809. URL: http://books.google.de/books?id=ri1muolw_YwC.
- Herrington, J. (2003). *Code Generation in Action*. In Action Series. Manning. ISBN: 9781930110977. URL: <http://books.google.de/books?id=VHVC8WnSgbYC>.
- Kelly, S. und J.P. Tolvanen (2008). *Domain-Specific Modeling: Enabling Full Code Generation*. Wiley. ISBN: 9780470249253. URL: http://books.google.de/books?id=GFFtRFkuU_AC.
- Klar, M. und S. Klar (2006). *Einfach generieren: Generative Programmierung verständlich und praxisnah*. Hanser Fachbuchverlag. ISBN: 9783446404489. URL: <http://books.google.de/books?id=6LS70wAACAAJ>.