



SPRECHEN SIE <http://cdn07.mydomain.cc/9rf7e2/i?HXID=iaj34jgt30qa3&qry=f#loaddelay?>

Auch und gerade für dynamische, komplexe Webseiten sind einfache, eindeutige und sprechende URLs ein wichtiger Faktor für die Benutzerfreundlichkeit.

| von JOHN LOUTZENHISER
Übersetzt von Dr. Andreas Zamperoni

Mit dem Aufkommen immer ausgeklügelter Web-Frameworks in den letzten zwei Jahrzehnten wurden viele Aspekte der Web-Entwicklung einfacher und eleganter. Andere Aspekte jedoch gestalten sich jetzt komplizierter und schwerfälliger. Beispielsweise war es in den frühen Tagen des world wide web mit ausschließlich statischen Webseiten ein Leichtes, einfache, sinnvolle – sprechende – URLs einzusetzen. Doch je komplexer die Technologien wurden, je mehr dynamische Contentgenerierung und Web-Applikationen statische Webseiten verdrängten – sie sind heute quasi nicht mehr anzutreffen –, desto stärker hat das Konzept einer eindeutigen, sprechenden URL gelitten.

REST VS. RESTFUL URL

REST bezeichnet ursprünglich ein Programmier- beziehungsweise Kommunikationsparadigma, insbesondere für verteilte Systeme auf Basis von Webservices. Im Zusammenhang mit Webadressen (URLs) greift „RESTful URLs“ eines der Prinzipien von REST auf: das der „selbsterklärenden“ Nachrichten, in diesem Falle interpretiert als „sprechende“ URLs.

URLs sprechend, einfach, eindeutig und beständig (das heißt lesezeichentauglich) – kurz gesagt: „RESTful“ – zu halten, ist vor allem dann eine Herausforderung, wenn ein Web-Framework verwendet wird, das auf dem Einzelseiten-Webanwendungsprinzip („Single Page Application“ [SPA]) basiert. Auch wenn es alles andere als trivial zu implementieren ist, ist es von großem Vorteil, URLs „RESTful“ zu halten, wenn man SPA-Frameworks verwendet.

Dieser Artikel zeigt, warum es sowohl aus der Sicht des Endnutzers als auch zur Suchmaschinenoptimierung („Search Engine Optimization“ [SEO]) sinnvoll ist, in die „RESTfulness“ von URLs zu investieren. Und er zeigt einige technische Möglichkeiten zur Umsetzung von RESTful URLs bei Einzelseiten-Webanwendungen auf.

DIE URL ALS „BENUTZERERFAHRUNG“

Der vielleicht wichtigste Grund für sprechende, intuitiv verständliche (RESTful) URLs ist, dass URLs eine wichtige Rolle beim sogenannten Nutzererlebnis einer Website („User Experience“ [UX]) spielen. Bereits 2007 wurde in einer Microsoft Research-Studie (Cutrell and Goan) festgestellt, wie wichtig URLs



Abbildung 1: Suchmaschinenergebnisse bestehen in der Regel aus einem Titel, der URL und einem Snippet (Teaser)

für Nutzer und für das Ansteuern von Webseiten sind. In dieser Eye-Tracking- und Nutzerfreundlichkeitsstudie haben Nutzer in Suchergebnissen die relative Wichtigkeit von (Webseiten-) Titel, Teaser und URL beim Lösen gestellter Navigations- und Informationsaufgaben bewertet. Zusätzlich wurden die Augenbewegungen erfasst, um zu ermitteln, wie viel Zeit die Nutzer mit dem Lesen des Titels, des Teasers und der URL verbrachten (siehe Abbildung 2). Die Ergebnisse zeigten, dass die URL in der Tat ein wichtiger Bestandteil für die Bewertung der Qualität (Relevanz) von Suchergebnissen ist.

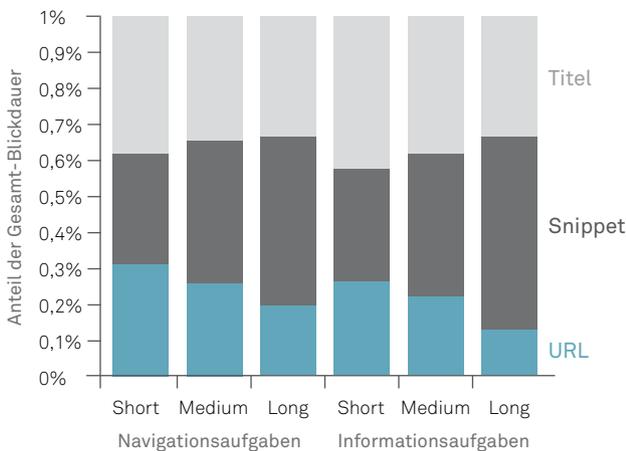


Abbildung 2: URLs sind ein wichtiger Bestandteil beim Ansteuern von Webseiten

URLs tragen zu einem positiven Nutzererlebnis bei, wenn sie RESTful sind. RESTful URLs, wie zum Beispiel <http://alletiere.de/slideshows/niedliche-katzen>, haben die folgenden Eigenschaften, die kryptische, verwirrende, „schmutzige“ URLs, wie zum Beispiel <http://cdn07.irgendeinedomäne.cc/9rf7e2/i?HXID=iaj34089jgt30ghqa3&qry=f#loaddelay>, nicht haben:

1. RESTful URLs geben eine eindeutige Antwort auf die Frage: „Wo befindet sich die (anzusteuern) Webseite?“ Titel und Teaser eines Suchergebnisses beantworten eher die Frage „Worum geht es auf der Webseite?“. Durch einen logischen und hierarchischen Aufbau (analog zu einer Ordnerstruktur) verweist eine sprechende URL auf eine eindeutig identifizierbare Lokation in den unendlichen Weiten des Internets.
2. RESTful URLs bleiben im Gedächtnis des Nutzers haften und sind leicht (wieder-)erkennbar. Das ist nützlich, wenn der Nutzer nach zuvor besuchten Webseiten sucht. Wenn er weiß, „wo“ sich eine Webseite befindet, weil die Adresse kurz, einprägsam, sprechend und eindeutig ist, findet er sie leichter wieder.
3. RESTful URLs sind vertrauenswürdig. In den oben genannten Beispielen wird der Nutzer die erste URL so wahrnehmen, dass sie verlässlich Informationen (Bilder) über niedliche Katzen enthalten wird, während es bei der zweiten URL keine Anhaltspunkte dafür gibt. Beim zweiten Beispiel fragt sich der Nutzer wahrscheinlich, warum die URL-Adresse so verschleiert wurde und ob sich möglicherweise etwas Bösartiges oder Zwielfichtiges hinter dem Link verbirgt.
4. RESTful URLs laden den Nutzer dazu ein, eine Website selbstständig zu „erforschen“ und auf ihr zu navigieren. Aus dem Aufbau der URL-Adressen kann man den Ort weiterer Informationen extrapolieren. Wo würde man als geneigter Nutzer im nachfolgenden Beispiel <http://dieuniversität.de/mathe/mueller/vorlesungsskripte.html> die Vorlesungsskripte der Physikvorlesungen von Prof. Heinrich vermuten?

SUCHMASCHINENOPTIMIERUNG UND RESTFUL URLS

URLs mit komplexen Parametern, wie zum Beispiel Session-IDs, Filter- und Sortierkriterien, Zähler oder Verweisinformationen (sogenannte referral ids), können die Suchmaschinen-Crawler



Abbildung 3: Bei SPAs werden alle Inhalte zustandsgesteuert auf einer Webseite dargestellt

vor Probleme stellen und zu schlechten Ranking- oder Indizierungsergebnissen führen.

Im Beispiel <http://www.derversand.de/produkte/produkte.asp?N=200063&Ne=500955&ref=foobar&Cn=zubehoer> steckt hinter den URL-Parametern eine kombinatorische Explosion potenzieller (zu durchsuchender) Webseiten, die sich hinter der „Single Page“ <http://www.derversand.de/produkte/produkte.asp> verbergen. Aus diesem Grund empfiehlt es sich für Website-Administratoren, besonders die URLs von Seiten, die von Crawlern durchsucht werden, frei von abstrakten, „nichtsprachlichen“ oder technischen Parametern zu halten, die den Inhalt der Webseiten verschleiern.

Allerdings spielt, auch wenn Suchmaschinen-Crawler ihre Probleme mit parametrisierten URLs haben, dieser Umstand in der Regel bei der Indizierung und dem Ranking einer Seite keine Rolle, da natürlich-sprachliche Begriffe in den URLs nicht berücksichtigt werden. Für einen Web-Crawler wären daher die beiden folgenden URLs gleich „gut“: <http://alletiere.de/slideshows/niedliche-katzen> oder <http://cdn07.cc/9rf7e2/cbj42>.

Und doch ist die erste URL aufgrund ihrer Lesbarkeit klar im Vorteil. Denn solche RESTful URLs werden häufiger angeklickt, kopiert und referenziert als kryptische, unverständliche URLs – eine indirekte Suchmaschinenoptimierung.

SINGLE PAGE APPLICATIONS (SPA)

Das Einzelseiten-Webanwendungsprinzip („Single Page Application“ [SPA]) zur Strukturierung und zur technischen Umsetzung von Websites erfreut sich in letzter Zeit immer größerer Beliebtheit. Eine SPA unterscheidet sich von traditionellen seitenbasierten Websites dadurch, dass sie nicht aus vielen einzelnen Seiten besteht, sondern – wie der Name es schon sagt – aus einer einzelnen Webseite, die verschiedene Zustände annehmen kann (siehe Abbildung 3).

Bei SPAs agiert eine einzelne Webseite als eine Art „Master“ für alle Inhalte einer Seite oder Webapplikation. Inhalte werden je nach nutzer- oder serverausgelöster Zustandsänderung (zum Beispiel durch Klicken von Links auf der Seite) automatisch in die entsprechenden Anzeigebereiche der Masterseite geladen. SPAs wurden ursprünglich für die Implementierung von Web-Anwendungen entwickelt, die Desktop-Anwendungen mit ihren Knöpfen und Ordnerstrukturen imitieren sollten. Zunehmend werden sie aber auch verwendet, um herkömmliche, informationsbasierte Websites umzusetzen, denn sie bieten dem Nutzer viele Vorteile, wie beispielsweise ein flüssigeres Nutzererlebnis durch Reduzierung bandbreitenbedingter Latenzzeiten. Gleichzeitig aber bedeuten sie eine größere Herausforderung bezüglich lesbarer, verständlicher URLs.

SPA-Technologien können grob in zwei verschiedene Kategorien eingeteilt werden:

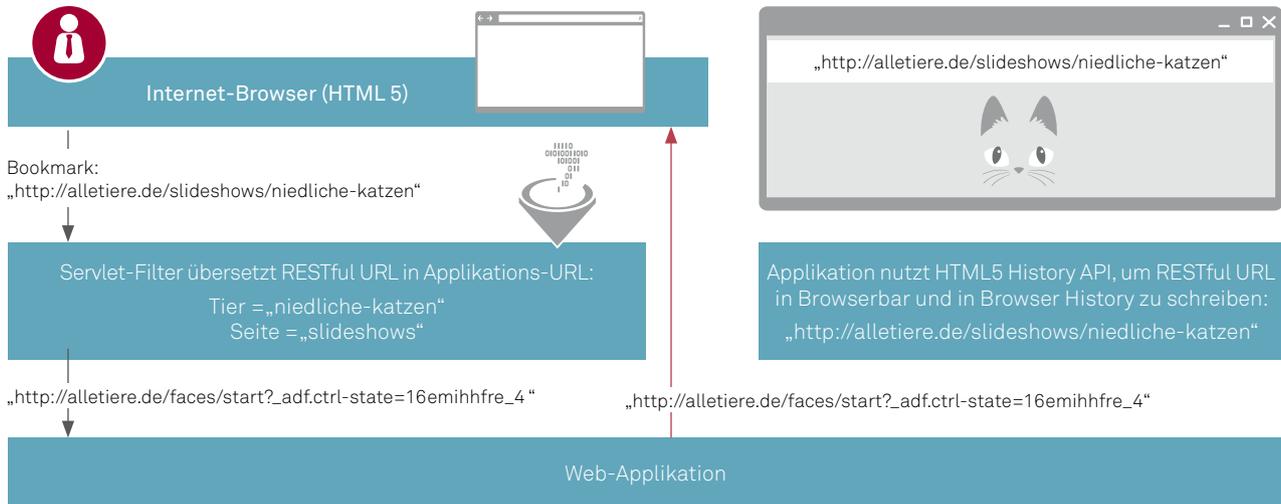


Abbildung 4 : Die Um- und Rückwandlung von RESTful URLs in Web-Applikationen

1. Webframeworks, die die Nutzerzustände (also das, was auf der Webseite angezeigt werden soll) im Client (Browser) managen; dazu gehören auch die so populären Javascript-Frameworks wie AngularJS und Ember.js. Client-seitig generierte URLs (zum Beispiel mittels AngularJS) sehen in der Regel wie folgt aus: ***http://alletiere.de/start/#/katzen***

Der erste Teil der URL bis zum „#“ („Anker“) verweist auf die Basisseite, die zustandsabhängig verändert werden kann. Der zweite Teil, nach dem „#“, beschreibt den „Seitenzustand“, also den anzuzeigenden Inhalt. Er wird durch den im Browser laufenden Frameworkcode interpretiert und als AJAX-Request mit der Anfrage, den entsprechenden Inhalt zu übermitteln, an den Server geschickt.

2. Webframeworks, die die Nutzerzustände auf dem Web- oder Applikationsserver managen; dazu gehören zum Beispiel JSF-basierte SPA-Frameworks wie Oracle-ADF und ADF-Faces. URLs, die durch ein JSF-basiertes SPA-Framework generiert werden, sind im Allgemeinen komplexer und technischer, das heißt für den menschlichen Leser undurchsichtiger und unverständlicher, wie im folgenden typischen Oracle-ADF-Beispiel: ***http://alletiere.de/faces/start?_adf.ctrl-state=16emihfre_4***

Der erste Teil der URL bis zum „?“ verweist auf die Basisseite, die zustandsabhängig verändert werden kann. Der zweite Teil, nach dem „?“, beschreibt den Seitenzustand. Allerdings verweist die-

ser zweite Teil auf eine Datenstruktur im Server, in der der Seitenzustand gesteuert wird. Der Zustand selbst lässt sich nicht, auch nicht verschlüsselt, aus der URL ablesen. Bei Änderungen des Zustandes, das heißt der Anzeige neuer Seiteninhalte, ändert sich die URL nicht notwendigerweise.

SUCHMASCHINENOPTIMIERUNG IN SPAS

Diese in SPA-Frameworks verwendeten URL-Strukturen haben in der Vergangenheit große Einschränkungen für die Nutzerfreundlichkeit und Suchmaschinenoptimierung nach sich gezogen.

Vor Oktober 2015 bedeuteten anker-basierte („#“) URLs signifikante Herausforderungen für Suchmaschinen wie Google (und andere). Denn bis zu diesem Zeitpunkt waren Googles Webcrawler noch nicht in der Lage, von besuchten Seiten heruntergeladenen Javascript-Code so gut auszuführen, wie es die modernen Web-Browser heute können. Wenn Googles Webcrawler auf URLs mit Ankersymbolen traf, verfügte er noch nicht über adäquate Mechanismen, die hinter den Ankersymbolen „versteckten“, zustandsbasierten Inhalte dynamisch nachzuladen. Diese Inhalte waren damit für Google weitgehend „unsichtbar“.

Um dieses Problem zu lösen, hatte Google sogar eine eigene HTML-Erweiterung, die sogenannte Google AJAX Crawling Specification, entwickelt. Sie sah vor, dass alle über die Ankernotationen darstellbaren Inhalte zusätzlich auch statisch, über

„normale“ URLs, erreichbar sein mussten. Diese „Lösung“ – eigentlich ein Umweg! – stellte die Websiteentwickler vor die aufwendige (und redundante Aufgabe), sicherzustellen, dass jeglicher dynamisch erzeugbare Content, um für Google erreichbar zu sein, zusätzlich auch statisch abgespeichert und adressierbar sein musste. Es entstanden Dienstleister (wie zum Beispiel Prerenderer.io), die gegen Gebühr Websites durchforsteten, dynamische Inhalte zusätzlich als statische Webseiten abspeicherten und Google „zuführten“.

Glücklicherweise hat Google im Oktober 2015 diese Lücke geschlossen und seinen Webcrawler um eine Javascript-Engine erweitert, die im Wesentlichen das Verhalten moderner Web-Browser nachstellen und dynamische Inhalte laden und indizieren kann. Diese Weiterentwicklung machte die Google AJAX Crawling Specification (und Dienste wie Prerenderer.io) im Grunde genommen obsolet.

Suchmaschinenoptimierung bei serverseitigen Technologien, wie im obigen ADF-Beispiel, stellen die Suchmaschinenanbieter derzeit immer noch vor größere Probleme. Technologien, bei denen Statusänderungen, das heißt Änderungen des angezeigten Inhalts, nicht mit in einer Änderung der URL einhergehen, weil sie im „Inneren“ des Webapplikationsservers abgehandelt werden, sind in Bezug auf eine Suchmaschinenoptimierung der Worst Case. Diese URLs sind kryptisch und für den Nutzer unverständlich, gleichzeitig sind die vielfältigen Inhalte, die sich dahinter verbergen, für Suchmaschinen unauffindbar.

EINE RESTFUL-LÖSUNG FÜR SERVERSEITIG GESTEUERTE SPAS

Doch wie können bei serverseitig gesteuerten SPAs, in diesem Fall einer Oracle-ADF-Anwendung, RESTful URLs umgesetzt werden? Zur Erinnerung: URLs, die durch diese Webtechnologie erzeugt und verarbeitet werden, http://alletiere.de/faces/start?_adf.ctrl-state=16emihhfre_4, sind kryptisch und ändern sich nicht notwendigerweise in der Anzeige des Webbrowsers, wenn der Nutzer auf der Seite navigiert.

Zunächst muss für die entsprechende Seite ein Seitenschema (RESTful scheme“) entworfen werden, das alle angezeigten Seiten in eine semantisch intuitive Struktur einordnet, wie zum Beispiel:
<http://alletiere.de/slideshows/niedliche-katzen>
<http://alletiere.de/slideshows/niedliche-hunde>
...
<http://alletiere.de/pflege/niedliche-katzen>
...

Dann gilt es, den Web-Browser so zu „täuschen“, dass er statt der durch die Webapplikation tatsächlich erzeugte URL die RESTful URL anzeigt. Mit der 2014 eingeführten HTML5-Javascript-API ist es möglich, die im Web-Browser angezeigte Adresse zu ändern. Der Web-Browser steuert zwar die Webseite mit der ADF-URL an, zeigt aber die RESTful-Adresse im Adressfeld an (siehe Abbildung 4, rechts)

Die HTML5-Lösung ist aber nur die halbe Miete, denn was, wenn der Nutzer die RESTful URL abspeichert oder weiterleitet? Die Webapplikation muss in der Lage sein, auch solche „nur angezeigten“ (RESTful) URLs zu verarbeiten.

Die Lösung liegt in der Implementierung eines Filters (eines sogenannten servlet filters), der die eingehenden URL-Anfragen abfängt und eingehende RESTful URLs in solche „übersetzt“, die die ADF-Webapplikation verarbeiten kann (siehe Abbildung 4, links).

ZUSAMMENFASSUNG

„Sprechende“, intuitiv verständliche (RESTful) URLs sind ein wichtiger Faktor für die Nutzerfreundlichkeit von Websites. Sie erlauben es den Nutzern, sich selbstständig(er) auf einer Seite zu bewegen, können leichter gemerkt werden und werden eher weiterverteilt.

RESTful URLs sind außerdem ein wichtiger Beitrag zur Suchmaschinenoptimierung, da sie direkt und indirekt (durch häufigeres Ansteuern) das Ranking verbessern. Sie werden in Ergebnislisten besser wahrgenommen und wirken vertrauenswürdiger.

Obwohl viele Webframeworks RESTful URLs nicht unterstützen, kann können sie doch mit einfachen oder zumindest überschaubaren Mitteln umgesetzt werden. ●

ANSPRECHPARTNER – JOHN LOUTZENHISER

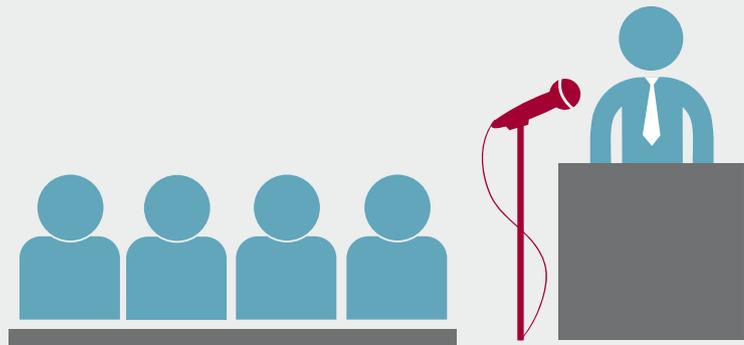
Senior IT Consultant

Public Sector Solutions Consulting

- +49 173 859 4235
- john.loutzenhiser@msg-systems.com



Veranstaltungshinweis



PLANEN SIE SCHON JETZT IHRE
WEITERBILDUNG FÜR 2017!

Vergabeverfahren und Vertragsgestaltung in agilen IT-Projekten der öffentlichen Hand

27.04.2017, München

13.09.2017, Bonn

Referenten: Werner Achtert, Leiter Public Sector Business Consulting, msg systems ag

Günther Pinkenburg, Rechtsanwalt und Geschäftsführender Gesellschafter MAYBURG Rechtsanwaltsgesellschaft mbH

IT-Management in der öffentlichen Verwaltung

16.–17.05.2017, München

15.–16.11.2017, Berlin

Referent: Werner Achtert, Leiter Public Sector Business Consulting, msg systems ag

Agile Methoden zum Management von IT-Projekten

27.–28.06.2017, München

06.–07.12.2017, Berlin

Referent: Werner Achtert, Leiter IT-Consulting Public Sector, msg systems ag

Weitere Informationen unter: www.fuehrungskraefte-forum.de