

ATURL: Actually Tiny URLs

Kaylee Frye, Derrial Book

2018-05-14

1. Einleitung

Das Universum braucht einen besseren URL Shortener.

URL Shortening erzeugt kurze URLs, die nur Weiterleitungen sind zu sehr viel längeren URLs. Das ist nützlich, weil Kurz-URLs besser zu merken sind und leichter mündlich weiterzugeben. Ebenfalls nützlich sind sie, wenn URLs im Text auftauchen (bspw. IRC).

Eine URL besteht aus mehreren Komponenten:

http : //web.io/ index? argument=somevalue #theAnchor
protocol host path parameter fragment

Für Kurz-URLs ist der path (engl. Pfad) relevant. Das Protokoll ist HTTP oder HTTPS, abhängig vom Besucher. Die Wahl des host-Namens ist nicht Teil der Softwareentwicklung, sondern des Betreibers. Parameter und Fragment werden nicht genutzt, da es mindestens einen zusätzlichen Buchstaben braucht und damit die URL verlängert.

Im weiteren Text nehmen wir an, dass der Dienst unter dem Hostnamen `atu.r1` angeboten wird.

2. Kriterien

2.1. Muss

M1 **Beschränkte Länge** NICHT IMPLEMENTIERT

Der Pfad von Kurz-URLs ist exakt 3 ASCII Buchstaben oder Ziffern lang. Das beschränkt die Anzahl der möglichen URLs auf knapp 200.000 Kurz-URLs.

M2 **Schnelle Weiterleitung Kurz- zu Lang-URL**

Implementiert durch: F1

M3 **Authentifizieren mit E-Mail oder Facebook**

Implementiert durch: F2

M4 **Rechtliche Vorgaben werden eingehalten**

Implementiert durch: F3 F4

Der Dienst befolgt alle in Deutschland geltenden Richtlinien, insbesondere das Telemediengesetz.

2.2. Kann

K1 **Authentifizieren mit Github**

Implementiert durch: F2

K2 **Seite mit Betreiberinfo** keine entsprechende Anforderung

Der Dienst bietet eine Seite „Über Uns“, mit Informationen zum Betreiber.

2.3. Abgrenzung

Keine Wahl Kurz-URL

A1

Ein Nutzer hat keine Möglichkeit die Auswahl einer Kurz-URL zu beeinflussen.

3. Produkteinsatz

Das Produkt soll mit minimalen Administrationskenntnissen zu betreiben sein.

Besucher des Diensts sollen diesen ohne Schulung oder andere Information benutzen können.

4. Produktumgebung

Das Programm soll als Servlet in einem Apache Tomcat betrieben werden.

Es stehen mindestens 2 AMD64 Kerne mit insgesamt 2GB shared RAM zur Verfügung.

5. Funktionale Anforderungen

F1 **Schnelle Weiterleitung** NICHT GETESTET
Implementiert: M2

Um zu einer Kurz- die Lang-URLs zu finden, benutzt der Dienst einen $O(1)$ Mechanismus. Es wird sichergestellt, dass mit einer zunehmenden Anzahl von Kurz-URLs im System das Finden nicht länger dauert.

F2 **Login-Möglichkeit auf Homepage**
Getestet durch: T1 Implementiert: M3 K1

Auf der Homepage <http://atu.r1/> sieht ein Besucher einen „Login via Facebook“ Knopf. Weitere Knöpfe wie „Login via Github“ sind möglich. Siehe Abb 1.

F3 **Auf jeder Seite ist ein Link „Impressum“**
Getestet durch: T2 Implementiert: M4

Entsprechend Telemediengesetz (TMG) §1 müssen gewisse Informationen über einen Betreiber jederzeit verfügbar sein. Die inzwischen etablierte Vorgehensweise ist auf allen (!) Seiten einen Link mit dem Text „Impressum“ anzubieten. Dieser Link führt auf eine Seite mit entsprechender Erklärung.

Auf jeder Seite ist ein Link „Datenschutz“

F4

Getestet durch: T2 Implementiert: M4

Entsprechend Telemediengesetz (TMG) §13 muss der Betreiber den Nutzer Speicherung und Verarbeitung seiner personenbezogenen Daten unterrichten. Die inzwischen etablierte Vorgehensweise ist auf allen (!) Seiten einen Link mit dem Text „Datenschutz“ anzubieten. Dieser Link führt auf eine Seite mit entsprechender Erklärung.

Daten werden persistent gespeichert NICHT GETESTET

F5

Implementiert:

Ein geordneter Neustart des Diensts führt nicht zu Datenverlust.

6. Nicht-Funktionale Anforderungen

Modernes Design

N1

Das Design soll modern und seriös wirken.

Persistenz

N2

Sollten in Zukunft Erweiterungen oder Updates notwendig werden, müssen die Daten (Kurz-URLs, E-Mailadressen) erhalten bleiben.

Erweiterbarkeit

N3

Das Produkt muss dahingehend erweiterbar sein, dass die Liste der E-Mail-URL Abbildung von authentifizierten Nutzern abgerufen werden kann. Wie das genau implementiert wird, ist nicht Teil dieses Projekts.

7. Tests

T1

Kurz-URL Erstellen

Testet: F2

T1.1 **Stand:** Besucher „Zoe Washburne“ hat einen Browser geöffnet.

Aktion: Zoe navigiert auf die Homepage `http://atu.r1/`.

Reaktion: Die Homepage wird angezeigt wie in Abb 1.

T1.2 **Stand:** Zoe hat einen Facebook-Account.

Aktion: Zoe drückt den „Login via Facebook“ Knopf.

Reaktion: Zoe wird eingeloggt und auf die Erstellseite wie in Abb 2 weitergeleitet.

T1.3 **Stand:**

Aktion: Zoe befüllt das Feld „URL“ mit `sehrlangedomain.com/undganzlangeURL.html` und drückt auf „Kurz-URL erstellen“.

Reaktion: Ihr wird eine Kurz-URL angezeigt wie z.B. `atu.r1/abc` wie in Abb 3. Statt „abc“, dürfen beliebige andere Buchstaben und Zahl angezeigt werden, allerdings exakt drei.

T1.4 **Stand:**

Aktion: Zoe navigiert auf die eben generierte Kurz-URL, z.B. `http://atu.r1/abc`.

Reaktion: Sie wird zu `sehrlangedomain.com/undganzlangeURL.html` weitergeleitet.

T2

Betreiberinfos lesen

Testet: F3 F4

T2.1 **Stand:** Besucher „Jayne Cobb“ ist auf der Homepage

Aktion: Er folgt dem Link mit dem Text „Datenschutz“

Reaktion: Ein Text mit allen Datenschutzinformationen wird ihm angezeigt.

T2.2 **Stand:**

Aktion: Jayne folgt dem Link mit dem Text „Impressum“

Reaktion: Ein Text mit Informationen des Betreibers wird ihm angezeigt.

A. Seitenentwürfe



Abbildung 1: Homepage mit Login-Funktion

Kurz-URL erstellen:

Gib eine lange URL in das folgende Feld ein und drücke auf den Knopf um eine Kurz-URL zu erstellen.

Lang-URL http://

Kurz-URL erstellen

[Über Uns](#) [Datenschutz](#)

Abbildung 2: Formular zur Generierung einer Kurz-URL. Getestet beispielsweise in T1 .

Kurz-URL erstellt

Gib eine lange URL in das folgende Feld ein und drücke auf den Knopf um eine Kurz-URL zu erstellen.

Lang-URL http://

Deine Kurz-URL: [atu.rl/f9v](#)

[Über Uns](#) [Datenschutz](#) [Impressum](#)

Abbildung 3: Anzeige der generierten Kurz-URL. Das Textfeld mit der Lang-URL kann nicht geändert werden.

B. Glossar

Besucher: Eine Person, welche den Dienst nutzt. Kann eingeloggt sein oder nicht.

Dienst: Die Software im laufenden Betrieb. Software as a Service.

Homepage: Seite, die beim Besuchen der Betreiberdomain *ohne Pfad* angezeigt wird. Auch „Startseite“.

Nutzer: Ein eingeloggter Besucher.