Uniform Resource Locator (URL's)

ID:	**TT-H5R**
Dokumenten URL:	**http://docs.tx7.de/TT-H5R**
Autor:	**Tom Gries <tom@tx7.de>**
Version:	**5.0.0 vom 01.07.2017**

>> Themen

- Definition
- Aufbau einer URL (die Schemes HTTP und FTP)
- Die einzelnen Bestandteile einer URL
- Die Schemes MAILTO und TEL
- Parameter zum Öffnen einer PDF-Datei
- Erlaubte Zeichen und Kodierung
- Referenzen

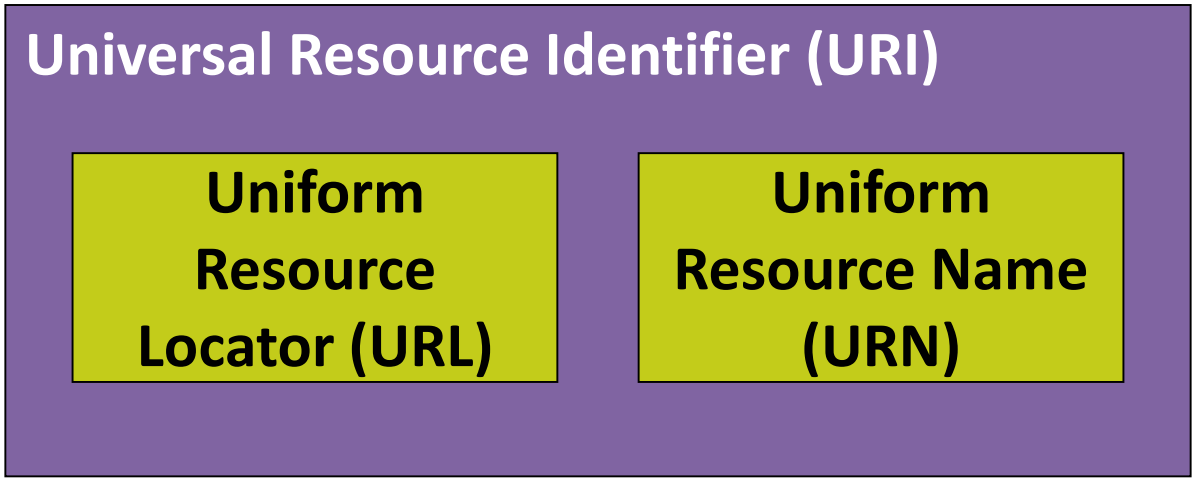
>> Definition

URL's sind die Adressen des Internet. Die Abkürzung URL steht hierbei für "Uniform Resource Locator". Das Wort "Uniform" zeigt an, dass hier eine gewisse Gleichheit - auch bei unterschiedlichen Ressourcen besteht.

Eine URL ist eine Unterform der URI's, also der "Universal Resource Identifier". Eine weitere Unterform der URI's ist der "Uniform Resource Name" (URN).

>> Definition

Visualisierung des Zusammenhangs zwischen URI, URL und URN:



Die Unterscheidung URI und URL/URN kam erst in RFC 2396 hinzu. Im ersten RFC 1738 von 1994 wurde noch keine Unterscheidung getroffen.

>> Definition

URI ist der Oberbegriff und benennt eine im Internet vorhandene Ressource:

- URL: Der Ort der Ressource kann bestimmt werden.
- URN: Der Ort der Ressource ist nicht (eindeutig) bestimmbar. Beispiel: ISBN (urn:isbn:3-527-70399-3). Der Zugriff erfolgt über den Namen.

>> Aufbau einer URL

Eine URI (und damit auch eine URL) hat folgenden Standardaufbau:

<scheme>:<scheme-specific-part>

Der <scheme-specific-part> ist immer abhängig von dem verwendetem Scheme. Das Scheme ist bei einer URL oft ein Protokoll, z. B. HTTP(S) oder FTP.

>> Aufbau einer URL

Schemes, die kein Protokoll repräsentieren sind zum Beispiel:

- MAILTO,
- TEL und
- FILE

Für URL Schemes, die eine hierarchische Beziehung repräsentieren, beginnt <scheme-specific-part> mit zwei Slashen "//". Die URL hat dann die Form

<scheme>://<authority><path>?<query>

>> Aufbau einer URL

Dabei setzt sich <authority> aus folgenden Komponenten zusammen:

<authority>:= <user>:<password>@<host>:<port>

Auch <path> besteht aus mehreren Bestandteilen. Diese sind vom Scheme abhängig. Für HTTP ist dies

<path>:= <url-path>[?<searchstring>][#<fragment>]

>> Aufbau einer URL

Für FTP sieht <path> wie folgt aus:

<path>:= <url-path>[;type=<typecode>]

Einige Schemes kommen ohne Pfadangabe aus, zum Beispiel MAILTO. Dieses Scheme hat den Aufbau:

mailto:<mailto-specific-part>

>> Aufbau einer URL

Komplette (absolute) URL‘s für die Protokolle HTTP(S) und FTP sehen wie folgt aus (hier zur Übersicht jeweils auf 2 Zeilen dargestellt):

http://[<user>:<password>@]<host>[:<port>]
/<url-path>[?<searchstring>][#<fragment>]

ftp://[<user>:<password>@]<host>[:<port>]
/<url-path>[;type=<typecode>]

>> Die einzelnen Bestandteile einer URL

<scheme>://[<user>:<password>@]<host>[:<port>]

Das Scheme kennzeichnet üblicherweise ein Internet Protokoll. Am Scheme erkennt man also, welche "Sprache" gesprochen werden soll.

Das Scheme ist ein Pflichtbestandteil einer absoluten URL. Es muss immer mit angegeben werden. Die meisten Browser fügen das Scheme "http" als Default automatisch ein, wenn dies in der Adresszeile weggelassen wird.

>> Die einzelnen Bestandteile einer URL

<scheme>://**[<user>:<password>@]**<host>[:<port>]

Username und Passwort sind optional. Es kann auch nur <user> angegeben werden. Dann ist das Passwort leer. Ein leerer Username oder ein leeres Passwort ist nicht das Gleiche wie kein Username oder kein Passwort.

- http://foo:bar@tx7.de (Username „foo“ und Passwort „bar“)
- http://foo@tx7.de (Username „foo“ und leeres Passwort)
- http://tx7.de (Kein Username und kein Passwort)

>> Die einzelnen Bestandteile einer URL

<scheme>://[<user>:<password>@]**<host>**[:<port>]

Der Host Teil zeigt letztendlich auf einen bestimmten Server im Internet. Dieser Server ist bestimmbar, aber nicht unbedingt vom User (Beispiel „Load Balancing“).

Notiert wird der Host Teil entweder als **Domainname** oder als **IP Adresse**. Bei der Schreibweise als IP Adresse gelten alle für IP Adressen erlaubten Schreibweisen (octetted, dezimal, octal, hexadezimal).

>> Die einzelnen Bestandteile einer URL

<scheme>://[<user>:<password>@]<host>**[:<port>]**

Der Port bezeichnet die Schnittstelle, auf der der Server für das angegebene Protokoll Anfragen entgegennimmt. Für einige Protokolle gibt es Standard Ports, z. B.:

Protokoll:	HTTP	HTTPS	FTP	SSH
Port:	80	443	21	22

Die Angabe des Ports ist optional. Wenn sie weggelassen wird, wird der für das verwendete Protokoll definierte Standardport verwendet.

>> Die einzelnen Bestandteile einer URL

/<url-path>[?<searchstring>][#<fragment>]

Für HTTP: Der Pfad inklusive Dateiname auf dem Zielserver. Diese Angabe reicht im Allgemeinen aus, um ein bestimmtes Dokument auszuliefern.

/<url-path>[;type=<typecode>]

Für FTP: Wie bei HTTP der Pfad inklusive Dateiname auf dem Zielserver.

>> Die einzelnen Bestandteile einer URL

/<url-path>**[?<searchstring>]**[#<fragment>]

Der Searchstring besteht aus einem oder mehreren Parametern. Die Parameter haben den Aufbau

Variable=Wert

Mehrere Parameter werden mit einem "&" voneinander getrennt. Der Searchstring selbst wird mit einem "?" eingeleitet. Das "=" trennt die Variable von seinem Wert.

>> Die einzelnen Bestandteile einer URL

/<url-path>**[?<searchstring>]**[#<fragment>]

Ein Searchstring sieht zum Beispiel wie folgt aus:

?var1=wert1&var2=wert2&var3=wert3

Der Searchstring ist optional. Er wird nur bei dynamischen Seiten benötigt. Bei statischen HTML Seiten hat der Searchstring keine Funktion und wird ignoriert.

>> Die einzelnen Bestandteile einer URL

/<url-path>[?<searchstring>]**[#<fragment>]**

Das Fragment (bzw. der Anker) verweist innerhalb eines Dokumentes auf eine bestimmte Stelle. Daher wird auch beim Ändern bzw. Anhängen des Fragments das Dokument nicht erneut vom Server abgerufen. Der Webserver hat also keine Kenntnis von diesem Fragment.

Die Auswertung wird nur Client-Seitig vorgenommen. Scriptsprachen, wie zum Beispiel PHP, können das Fragment daher nicht auswerten. Javascript aber schon.

>> Beispiele für HTTP

Normale HTTP URL (nur Host und Pfad):

http://tx7.de/index.htm

Inklusive Username, Passwort und Port:

http://tux:frikadelle@scripte.tx7.de:8080/

Inklusive Query-String:

http://www.google.de/search?q=2%5E10%2B1

Inklusive Fragment:

http://de.wikipedia.org/wiki/Uniform_Resource_Locator#fragment

>> Beipiele für FTP

Normale FTP URL (nur Host):

ftp://ftp.tx7.de

Inklusive Username, Passwort und Port:

ftp://tux:frikadelle@scripte.tx7.de:2121/

Inklusive Pfad:

ftp://ftp.uni-kl.de/pub/linux/fedora/linux

Inklusive Typecode:

ftp://ftp.uni-kl.de/pub/linux/fedora/linux;type=i

>> Das Scheme MAILTO

Das Scheme MAILTO hat keine Hierarchie und wird daher ohne die beiden Slashe „//“ notiert. Auch fehlen die anderen Bestandteile wie bei HTTP und FTP. Allgemein wird dieses Scheme wie folgt notiert:

mailto:<local-part@domain>[<mailstring>]

>> Das Scheme MAILTO

<local-part@domain>[<mailstring>]

Der local-part, also alles was links von dem @-Zeichen steht, ist die Mailbox auf dem Mailserver. Rechts davon steht der Domain Name.

Beispiel:

mailto:webmaster@tx7.de

Der local-part bezeichnet also wie die Pfadangabe bei HTTP und FTP eine Ressource auf dem Zielrechner.

>> Das Scheme MAILTO

<local-part@domain>**[<mailstring>]**

Der Mailstring hat vom Aufbau her Ähnlichkeiten mit dem Querystring bei HTTP. Allerdings sind die Namen der Variablen fest definiert. Diese Variablen sind zum Beispiel „subject“, „body“, „cc“ oder „bcc“.

Viele Clients interpretieren "subject" und "body" korrekt. Bestimmte Zeichen müssen aber kodiert werden wie Zum Beispiel ein Leerzeichen mit **%20** und ein Zeilenumbruch mit **%0D%0A**.

>> Beispiele für MAILTO

Mit Subject inkl. Leerzeichen:

mailto:webmaster@tx7.de?subject=Bitte%20um%20Rückruf

Mit Body inkl. Leerzeichen und Zeilenumbruch[1]:

mailto:webmaster@tx7.de?body=Hallo,%0D%0Abitte%20melden!

Mit Subject und Body:

mailto:info@tx7.de?subject=Verteiler&body=unsubscribe

[1] Hier immer %0D%0A. Ansonsten: Win = %0D%0A (CR+LF), UNIX/Linux = %0A (LF), Mac = %0D (CR)

>> Das Scheme TEL

<phone-number>[<dailstring>]

Ein relativ neues Scheme ist TEL. Mit diesem Scheme ist auf Mobiltelefonen die Telefon-App verbunden. Auf Laptops oder PC kann zum Beispiel Skype damit verknüpft sein.

Grundsätzlich ist die internationale Notation zu verwenden. Ausnahmen sollten nur Service-Nummern bilden, die nicht international notiert werden können, wie zum Beispiel 112.

>> Das Scheme TEL

<phone-number>[<dailstring>]

Eine internationale Telefonnummer beginnt immer mit einem '+', gefolgt von der Länderkennung. Visuelle Trennzeichen innerhalb der Nummer sind erlaubt. Dies sind die 4 Zeichen:

- . ( )

Beispiel: tel:+49--700-200-100

>> Das Scheme TEL

<phone-number>**[<dailstring>]**

Der Dialstring ist nützlich, um nach dem eigentlichen Verbindungsaufbau noch Informationen in Form von DTMF Tönen/Sequenzen zu senden. Die komplette Telefonnummer kann dann in den Kontakten eines mobilen Endgerätes gespeichert werden. Als Anker in einem HTML Dokument wird es nicht durchgängig unterstützt. Unter Android werden die nichtnumerischen Zeichen meist herausgefiltert (Stand Mitte 2017).

>> Beispiele für TEL

Beschreibung	Zeichen
1-2 Sek. Pause	, oder p
Auf Freizeichen warten	w
Auf Ende des Verbindungsaufbau warten	; oder ;postd=
DTMF Digits	0-9*#

Nach Verbindungsaufbau 2 - 4 Sekunden warten und 123 senden und mit # bestätigen:

+49--0700-0200-0100,,123#

Anruf über Calling-Card inkl. der PIN 987654:

+49--0700-0200-0100,987654#44-123-4567#

>> Parameter zum Öffnen einer PDF-Datei

Es ist zwar kein Internet-Standard, aber in einer URL für ein PDF-Dokument kann man - ähnlich wie beim Query-String - Parameter mit angeben. Allerdings können diese nicht mit einem ? eingeleitet werden, da dies vom Webserver ausgewertet werden würde.

Adobe hat daher das Fragment-Zeichen # zum Einleiten der Parameterliste gewählt, da dieses nur lokal verarbeitet wird.

>> Parameter zum Öffnen einer PDF-Datei

Mit den Parametern werden unterschiedliche Aktionen unterstützt, zum Beispiel das Springen zu einer bestimmten Stelle (benanntes Ziel, Seite, Kapitel, Kommentar) oder die initiale Darstellung der zuerst angezeigten Seite (z. B. Zoom und Seitendarstellung).

Auch das Anzeigen der Lesezeichen, Vorschau, Toolbar, Statusbar etc. kann mit Parametern gesteuert werden. Selbst das Suchen ist mit den Parametern möglich. Die komplette Beschreibung der Parameter ist unter http://docs.tx7.de/TT-U9M zu finden.

>> Parameter zum Öffnen einer PDF-Datei

/doc.pdf**[#<PDF-Parameterlist>]**

Mehrere Parameter können mit & oder # getrennt werden. Die Parameter werden von links nach rechts abgearbeitet. Daher müssen Seitenaktionen vor Zoomaktionen angegeben werden.

Der Parameterteil einer URL zum Öffnen einer PDF-Datei sieht - bis auf das einleitende # (statt ?) - wie folgt aus:

#var1=wert1&var2=wert2&var3=wert3

>> Parameterbeispiele für PDF-Dateien

/doc.pdf**[#<PDF-Parameterlist>]**

In dem Dokument nach dem Wort "URL" suchen:

http://tx7.de/doc.pdf#search="URL"

Zu Seite 77 springen:

http://tx7.de/doc.pdf#page=77

>> Parameterbeispiele für PDF-Dateien

/doc.pdf**[#<PDF-Parameterlist>]**

Das Dokument mit Lesezeichen auf Seite 2 öffnen:

http://tx7.de/doc.pdf#pagemode=bookmarks&page=2

Das Dokument in 80% Seitengröße öffnen:

http://tx7.de/doc.pdf#zoom=80

>> Erlaubte Zeichen in einer URL

In einer URL sind unkodiert nur die Zeichen erlaubt, die zu der Gruppe der Reservierten und der Nicht-Reservierten Zeichen gehören.

HEX	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOH	ENQ	ACK	BEL	BS	HAT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Reserviert: gen-delims

Reserviert: sub-delims

Nicht-Reserviert

Nicht erlaubt: unwise

Nicht erlaubt: Steuerzeichen

>> Erlaubte Zeichen in einer URL

Reservierte Zeichen müssen unter bestimmten Voraussetzungen URL-kodiert werden.

Auch die **Nicht-Reservierten** Zeichen können URL-kodiert werden. Es wird allerdings empfohlen, sie nicht zu kodieren.

Alle **anderen Zeichen** müssen **immer** URL-kodiert werden.

>> URL-Kodierung (percent-encoding)

Eine URL zu kodieren bedeutet ein oder alle Zeichen einer URL durch eine (transportsichere) Schreibweise zu ersetzen. Hierzu wird das zu kodierende Zeichen durch die hexadezimale Stellenbezeichnung im (ASCII) Zeichensatz ersetzt. Die Kodierung wird durch das %-Zeichen gekennzeichnet – gefolgt von 2 hexadezimalen Ziffern.

<u>Beispiel:</u>

A:= %41

➔Daher kann % in einer URL nie literal verwendet werden.

>> Beispiele für URL-Kodierung (percent-encoding)

http://tx7.de

http://%74%78%37%2E%64%65/

Eine Datei mit Leerzeichen (My Picture.jpg) auf dem Server tx7.de

http://tx7.de/My%20Picture.jpg

Das % literal im Subject eines Mailstrings

mailto:tom@tx7.de?subject= 20%25%20auf%20Alles

>> Referenzen

RFC 3986 – URI Generic Syntax:

http://docs.tx7.de/rfc3986

RFC 1738 - Uniform Resource Locator (URL) - beinhaltet auch FTP:

http://docs.tx7.de/rfc1738

RFC 7230 - The HTTP and HTTPS URL scheme:

http://docs.tx7.de/rfc7230 (Section 2.7)

RFC 6068 - The mailto URI scheme:

http://docs.tx7.de/rfc6068

RFC 3966 - The tel URI for Telephone Numbers:

http://docs.tx7.de/rfc3966

>> Referenzen

RFC 4967 – Dial String Parameter for SIP URI:

http://docs.tx7.de/rfc4967

IANA List of URI Schemes:

http://docs.tx7.de/TT-NTQ

Percent-Encoding:

http://docs.tx7.de/TT-ZGA

PDF Open Parameters:

http://docs.tx7.de/TT-U9M