A cheatsheet for the Constrained Application Protocol (CoAP)

Article · January 2013

CITATION
1

READS
325

1 author:

Markus Becker
Universität Bremen
75 PUBLICATIONS **367** CITATIONS

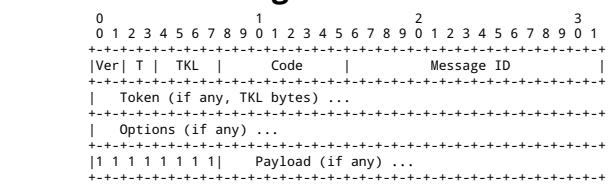
SEE PROFILE

Constrained Application Protocol

(RFC 6690, draft-ietf-core-coap-13, draft-ietf-core-block-10, draft-ietf-core-observe-07)

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks.

CoAP Message Format



Ver: Version, T: Type, TKL: Token Length

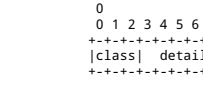
Method types

Type	Name
0	CONfirmable
1	NON-confirmable
2	ACKnowledgement
3	ReSeT

Method codes

Code	Name
1	GET
2	POST
3	PUT
4	DELETE

Response codes



Class	
2.xx	Success
4.xx	Client Error
5.xx	Server Error

Code	Description
65	2.01 Created
66	2.02 Deleted
67	2.03 Valid
68	2.04 Changed
69	2.05 Content
128	4.00 Bad Request
129	4.01 Unauthorized
130	4.02 Bad Option
131	4.03 Forbidden
132	4.04 Not Found
133	4.05 Method Not Allowed
134	4.06 Not Acceptable
140	4.12 Precondition Failed
141	4.13 Request Entity Too Large
143	4.15 Unsupported Content-Format
160	5.00 Internal Server Error
161	5.01 Not Implemented
162	5.02 Bad Gateway
163	5.03 Service Unavailable
164	5.04 Gateway Timeout
165	5.05 Proxying Not Supported

Options

No.	C	U	N	R	Name	Format	Length	Default
1	x			x	If-Match	opaque	0-8	(none)
3	x	x	-		Uri-Host	string	1-255	(see below)
4				x	ETag	opaque	1-8	(none)
5	x				If-None-Match	empty	0	(none)
7	x	x	-		Uri-Port	uint	0-2	(see below)
8				x	Location-Path	string	0-255	(none)
11	x	x	-	x	Uri-Path	string	0-255	(none)
12					Content-Format	uint	0-2	(none)
14		x	-		Max-Age	uint	0-4	60
15	x	x	-	x	Uri-Query	string	0-255	(none)
16				x	Accept	uint	0-2	(none)
20				x	Location-Query	string	0-255	(none)
35	x	x	-		Proxy-Uri	string	1-1034	(none)
39	x	x	-		Proxy-Scheme	string	1-255	(none)

C=Critical, U=Unsafe, N=No-Cache-Key, R=Repeatable

Content-Formats

Media type	Id.
text/plain;charset=utf-8	0
application/link-format	40
application/xml	41
application/octet-stream	42
application/exi	47
application/json	50

URI schemes

```
coap-URI = "coap:" "//" host [ ":" port ] path-abempty [ "?" query ]
coaps-URI = "coaps:" "//" host [ ":" port ] path-abempty [ "?" query ]
```

Transmission parameters

name	default value
ACK_TIMEOUT	2 seconds
ACK_RANDOM_FACTOR	1.5
MAX_RETRANSMIT	4
NSTART	1
DEFAULT_LEISURE	5 seconds
PROBING_RATE	1 Byte/second

Link Format .well-known/core

Link format can be used to describe hosted resources, their attributes, and other relationships between links.
Example:

```
REQ: GET /.well-known/core

RES: 2.05 Content

</sensors>;ct=40;title="Sensor Index",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<http://www.example.com/sensors/t123>;anchor="/sensors/temp";rel="describedby",
</t>;anchor="/sensors/temp";rel="alternate"
```

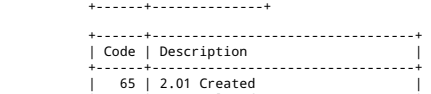
ABNF:

```
Link            = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value      = "<" URI-Reference ">" *( ";" link-param )
link-param      = ( ( "rel" "=" relation-types )
                / ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
                / ( "rev" "=" relation-types )
                / ( "hreflang" "=" Language-Tag )
                / ( "media" "=" ( MediaDesc
                      / ( DQUOTE MediaDesc DQUOTE ) ) )
                / ( "title" "=" quoted-string )
                / ( "title*" "=" ext-value )
                / ( "type" "=" ( media-type / quoted-mt ) )
                / ( "rt" "=" relation-types )
                / ( "if" "=" relation-types )
                / ( "sz" "=" cardinal )
                / ( link-extension ) )
link-extension = ( parmname [ "=" ( ptoken / quoted-string ) ] )
                / ( ext-name-star "=" ext-value )
ext-name-star  = parmname "*" ; reserved for RFC-2231-profiled
                              ; extensions.  Whitespace NOT
                              ; allowed in between.
ptoken         = 1*ptokenchar
ptokenchar     = "!" / "#" / "$" / "%" / "&" / "'" / "("
                / ")" / "*" / "+" / "-" / "." / "/" / DIGIT
                / ":" / "<" / "=" / ">" / "?" / "@" / ALPHA
                / "[" / "]" / "^" / "_" / "`" / "{" / "|"
                / "}" / "~"
media-type     = type-name "/" subtype-name
quoted-mt      = DQUOTE media-type DQUOTE
relation-types = relation-type
                / DQUOTE relation-type *( 1*SP relation-type ) DQUOTE
relation-type  = reg-rel-type / ext-rel-type
reg-rel-type   = LOALPHA *( LOALPHA / DIGIT / "." / "-" )
ext-rel-type   = URI
cardinal       = "0" / ( %x31-39 *DIGIT )
LOALPHA        = %x61-7A   ; a-z
quoted-string  = <defined in [RFC2616]>
URI            = <defined in [RFC3986]>
URI-Reference  = <defined in [RFC3986]>
type-name      = <defined in [RFC4288]>
subtype-name   = <defined in [RFC4288]>
MediaDesc      = <defined in [W3C.HTML.4.01]>
Language-Tag   = <defined in [RFC5646]>
ext-value      = <defined in [RFC5987]>
parmname       = <defined in [RFC5987]>
```

Block

In order to transfer larger payloads with CoAP — for instance, for firmware updates — the Block option can be used.

No.	C	U	N	R	Name	Format	Length	Default
23	x	x			Block2	uint	0-3 B	(none)
27	x	x			Block1	uint	0-3 B	(none)
28			x		Size	uint	0-4 B	(none)



Observe

In order to follow state changes of CoAP resources the Observe option can be used.

No.	C	U	N	R	Name	Format	Length	Default
6		x	x		Observe	empty/uint	0 B/0-3 B	(none)

References

This cheatsheet is based on and heavily stole from the following documents:

Link-format: http://tools.ietf.org/html/rfc6690
CoAP: http://tools.ietf.org/html/draft-ietf-core-coap-13
Block: http://tools.ietf.org/html/draft-ietf-core-block-10
Observe: http://tools.ietf.org/html/draft-ietf-core-observe-07