

VarnishQ - Referenz

Patrick Krusenotto

August 3, 2016

Contents

1	DSL-Makros	2
1.1	any (expr)	2
1.2	app (expr source)	2
1.3	cnt (n expr source)	3
1.4	objheader (name)	3
1.5	rxheader (name)	3
1.6	select (expr source)	3
1.7	txheader (name)	3
1.8	type-is (a)	3
1.9	where (expr source)	3
2	DSL-Funktionen	3
2.1	ban-list ()	3
2.2	bulk (source)	3
2.3	contains (string pattern)	3
2.4	dmp (source)	4
2.5	dump (source)	4
2.6	file (name) Erzeugt Pipe aus Logdatei	4
2.7	head (n l)	4
2.8	is-prefix (prefix string)	4
2.9	live ()	4
2.10	matches (pattern string)	4
2.11	max-age (r &optional (default nil))	4
2.12	nth-num (n string &optional (default 0))	4
2.13	numb (pattern string)	4
2.14	select-tag (tag source)	4
2.15	select-tags (tags source)	5

2.16	substr (s from &optional to)	5
2.17	tab (table)	5
2.18	tabcar (l)	5
2.19	take (n source)	5
3	interne Macros	5
3.1	aif (test then &optional else)	5
3.2	collect (symbols body)	6
4	interne Funktionen	6
4.1	code (x)	6
4.2	col-widths-of-table (table)	6
4.3	data (x) "datenfeld eines Satzes" (nth 3 x))	6
4.4	dissect (pattern string)	6
4.5	find-tag-data (tag req)	6
4.6	flatten-opnds (l)	6
4.7	inc-hash (hk htab)	6
4.8	iterator (s); Zeilen-Iterator eines Streams erstellen.	6
4.9	live-stream ()	6
4.10	logfile-stream (name)	6
4.11	parse (line)	6
4.12	print-log-line (l)	6
4.13	tag (x) "Tag eines Satzes" (nth 1 x))	6
4.14	thread (x) "thread-# eines Satzes" (nth 0 x))	6
4.15	time-string (time)	6
4.16	varnish-request-aggregator (source)	6

1 DSL-Makros

1.1 any (expr)

Wird innerhalb von Where verwendet, um auszudrücken, dass bei mehrfach-tags irgendeine Zeile eine bestimmte Eigenschaft haben muss

1.2 app (expr source)

Erstellt eine Pipe, die einen Bestimmten Ausdruck auf jeden Request einer anderen Pipe anwendet

1.3 cnt (n expr source)

Zählt die n nächsten Elemente einer Pipe anhand des Merkmals *expr*

1.4 objheader (name)

Liefert des Headerwert eines bestimmten Namens aus dem ObjHeader

1.5 rxheader (name)

Liefert des Headerwert eines bestimmten Namens aus dem RxHeader

1.6 select (expr source)

Liefert eine Pipe, die Zeilen nach einem Bestimmten Kriterium auswählt

1.7 txheader (name)

Liefert des Headerwert eines bestimmten Namens aus dem TxHeader

1.8 type-is (a)

Stellt fest, ob der aktuelle Request vom typ a ist. a kann 'c oder 'b sein.

1.9 where (expr source)

Liefert eine Pipe, die die aus der Pipe Source stammenden Objekte nach einem bestimmten Kriterium abfiltert.

2 DSL-Funktionen

2.1 ban-list ()

Liefert die aktuelle Ban-List des Caches

2.2 bulk (source)

Generiert Bulk-Ausgabe einer Pipe

2.3 contains (string pattern)

Stellt fest, ob pattern in string enthalten ist

2.4 dmp (source)

Liefert Taballarische Ausgabe aus einer Pipe

2.5 dump (source)

Liefert Dump einer Pipe

2.6 file (name) Erzeugt Pipe aus Logdatei

Liefert eine Pipe, die eine Datei einliest

2.7 head (n l)

Liefert die ersten n Elemente einer liste

2.8 is-prefix (prefix string)

Stellt fest, ob prefix ein Präfix zu string ist

2.9 live ()

Erzeugt Pipe aus *varnishlog*.

2.10 matches (pattern string)

Stellt fest, ob regex-pattern in einem string enthalten ist

2.11 max-age (r &optional (default nil))

Liefert das Max-Age eines Requests

2.12 nth-num (n string &optional (default 0))

Liefert die n-te Zahl innerhalb eines Strings

2.13 numb (pattern string)

Liefert die Zahl innerhalb eines Strings

2.14 select-tag (tag source)

Liefert eine Pipe, die ein bestimmtes Tag eines Requests passieren lässt

2.15 select-tags (tags source)

Liefert eine Pipe, die nur bestimmte Tags eines Requests passieren lässt

2.16 substr (s from &optional to)

Sichere Version von subseq

2.17 tab (table)

Gibt eine Liste tabellarisch aus

2.18 tabcar (l)

Gibt eine Liste tabellarisch aus

2.19 take (n source)

Entnimmt einer Pipe eine bestimmte Anzahl Requests

3 interne Macros

3.1 aif (test then &optional else)

"anaphorisches if" Anaphoric Macros numbers=left,basicstyle=,columns=fullflexible,keywordstyle
 ,caption= (aif (cdr x) (print it))

3.2 collect (symbols body)

4 interne Funktionen

4.1 code (x)

4.2 col-widths-of-table (table)

4.3 data (x) "datenfeld eines Satzes" (nth 3 x))

4.4 dissect (pattern string)

4.5 find-tag-data (tag req)

4.6 flatten-opnds (l)

4.7 inc-hash (hk htab)

4.8 iterator (s); Zeilen-Iterator eines Streams erstellen.

4.9 live-stream ()

4.10 logfile-stream (name)

4.11 parse (line)

4.12 print-log-line (l)

4.13 tag (x) "Tag eines Satzes" (nth 1 x))

4.14 thread (x) "thread-# eines Satzes" (nth 0 x))

4.15 time-string (time)

4.16 varnish-request-aggregator (source)