

NYTProf Performance Profile

◀ line view ▶

For /opt/rt4/sbin/rt-server

Run on Tue Mar 31 10:44:15 2015

Reported on Tue Mar 31 10:45:38 2015

Filename	/opt/rt4/share/html/REST/1.0/Forms/ticket/history
Statements	Executed 4103 statements in 28.8ms

Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
75	1	1	83.1ms	7.34s	HTML::Mason::Commands::_ANON_[:51] (recurses: max depth 1, inclusive time 7.15s)

Call graph for these subroutines as a [Graphviz dot language file](#).

Line	State ments	Time on line	Calls	Time in subs	Code
1					
--					
5					# spent 7.34s (83.1ms+7.26s) within HTML::Mason::Commands::_ANON_[:opt/rt4/share/html/REST/1.0/Forms/ticket/history:51] which was called 75 times, avg 97.9ms/call: # 75 times (83.1ms+7.26s) by HTML::Mason::Component::run at line 138 of HTML/Mason/Component.pm, avg 97.9ms/call
--					
12	75	22µs			
--					
18	75	61µs			
--					
51	75	62µs			\$id = \$_[\$pos{'id'}] ;
52	76	108µs	1	15µs	#line 52 "/opt/rt4/share/html/REST/1.0/Forms/ticket/history" # spent 15µs making 1 call to HTML::Mason::Component::new
53	75	74µs			\$args = exists \$pos{'args'} ? \$_[\$pos{'args'}] : undef;
54	75	105µs			#line 53 "/opt/rt4/share/html/REST/1.0/Forms/ticket/history" \$format = exists \$pos{'format'} ? \$_[\$pos{'format'}] : undef;
55					#line 54 "/opt/rt4/share/html/REST/1.0/Forms/ticket/history" \$fields = exists \$pos{'fields'} ? \$_[\$pos{'fields'}] : undef;
56					# spent 1.40ms making 75 calls to DBIx::SearchBuilder::Record::new, avg 19µs/call
57	75	256µs	75	1.40ms	
58	75	95µs			}
59					\$m->debug_hook(\$m->current_comp->path) if (HTML::Mason::Compiler::IN_PERL_DB());
60	75	117µs	75	18.3ms	# spent 18.3ms making 75 calls to RT::Ticket::Load, avg 244µs/call
61	75	80µs	75	35µs	#line 56 "/opt/rt4/share/html/REST/1.0/Forms/ticket/history" # spent 35µs making 75 calls to RT::Record::Id, avg 469ns/call
62					
63					my \$ticket = RT::Ticket->new(\$session{CurrentUser});
64					my (\$c, \$o, \$k, \$e) = ("", {}, {}, "");
65	75	146µs	75	78.2ms	# spent 78.2ms making 75 calls to RT::Ticket::Transactions, avg 1.04ms/call
66	75	145µs	75	522ms	\$ticket->Load(\$id); # spent 522ms making 75 calls to DBIx::SearchBuilder::Count, avg 6.96ms/call
67					unless (\$ticket->Id) {
68	75	76µs			return ["# Ticket \$id does not exist.", {}, {}, 1] ;
69					}
70					
71					my \$trans = \$ticket->Transactions();
72	1	600ns			my \$total = \$trans->Count();
73					
74					if (\$args) {
75	75	272µs			chomp \$args;
76	75	24µs			}
77	75	138µs			else {
78					\$args = '';
79					}
80					
81					my @arglist = split('/', \$args);
82	1	300ns			my (\$type, \$tid);
83					if (defined \$arglist[0] && \$arglist[0] eq 'type') {
84					\$type = \$arglist[1];
85	75	264µs	74	9.26ms	} elsif (defined \$arglist[0] && \$arglist[0] eq 'id') { # spent 9.26ms making 74 calls to RT::SearchBuilder::Limit, avg 125µs/call
86					\$tid = \$arglist[1];
87					} else {
88					\$type = \$args;
89					}
90					
91					if (\$type) {
92					# Create, Set, Status, Correspond, Comment, Give, Steal, Take, Told
93					# CustomField, AddLink, DeleteLink, AddWatcher, DelWatcher
94					if (\$args =~ /^links?\$/) {
95					\$trans->Limit(FIELD => 'Type', OPERATOR => 'LIKE', VALUE => '%Link');
96					}
97					elsif (\$args =~ /^watchers?\$/) {
98					\$trans->Limit(FIELD => 'Type', OPERATOR => 'LIKE', VALUE => '%Watcher');
99					}
100					else {
101	75	10.4ms	5624	1.50ms	\$trans->Limit(FIELD => 'Type', OPERATOR => '=', VALUE => \$type); # spent 1.50ms making 5624 calls to DBIx::SearchBuilder::Record::DESTROY, avg 267ns/call
102	74	12µs			}
103	74	569µs	74	6.08ms	} elsif (\$tid) { # spent 6.08ms making 74 calls to DBIx::SearchBuilder::Record::new, avg 82µs/call
104					\$trans->Limit(FIELD => 'Id', OPERATOR => '=', VALUE => \$tid);
105					}
106					
107	74	196µs	74	48.8ms	if (\$tid) { # spent 48.8ms making 74 calls to RT::Ticket::Transactions, avg 659µs/call
108	74	26µs			my @data;
109	74	7.35ms	11026	4.69s	my \$t = RT::Transaction->new(\$session{CurrentUser}); # spent 4.69s making 5550 calls to DBIx::SearchBuilder::Next, avg 844µs/call # spent 5.96ms making 5476 calls to RT::Record::Id, avg 1µs/call
110					
111					# this paragraph limits the transaction ID query to transactions on this ticket.
112	74	146µs	74	4.37ms	# Otherwise you can query any transaction from any ticket, which makes no sense. # spent 4.37ms making 74 calls to DBIx::SearchBuilder::Record::Load, avg 59µs/call
113					my \$transactions = \$ticket->Transactions;
114					my \$tok=0;
115					while (my \$T = \$transactions->Next()) {
116					\$tok=1 if (\$T->Id == \$tid)
117					}
118	74	173µs	74	63µs	if (\$tok) { # spent 63µs making 74 calls to RT::Record::Id, avg 853ns/call
119	74	237µs	74	2.38ms	\$t->Load(\$tid); # spent 2.38ms making 74 calls to RT::Transaction::Ticket, avg 32µs/call
120					} else {
121	74	213µs	74	56.8ms	return ["# Transaction \$tid is not related to Ticket \$id", {}, {}, 1] ; # spent 56.8ms making 73 calls to DBIx::SearchBuilder::Record::_ANON_[:DBIx/SearchBuilder/Record.pm:423], avg 778µs/call # spent 10µs making 1 call to DBIx::SearchBuilder::Record::AUTOLOAD
122					}
123	74	199µs	74	33.5ms	# spent 33.5ms making 73 calls to DBIx::SearchBuilder::Record::_ANON_[:DBIx/SearchBuilder/Record.pm:423], avg 459µs/call # spent 11µs making 1 call to DBIx::SearchBuilder::Record::AUTOLOAD

Line	State ments	Time on line	Calls	Time in subs	Code
124					push @data, [id => \$t->Id];
125	74	238µs	74	36.9ms	push @data, [Ticket => \$t->Ticket] # spent 36.8ms making 73 calls to DBIx::SearchBuilder::Record::__ANON_[DBIx/SearchBuilder/Record.pm:423], avg 505µs/call # spent 11µs making 1 call to DBIx::SearchBuilder::Record::AUTOLOAD
126					if (!%\$fields exists \$fields->{lc 'Ticket'});
127	74	251µs	74	36.6ms	push @data, [TimeTaken => \$t->TimeTaken] # spent 36.6ms making 74 calls to RT::Transaction::OldValue, avg 495µs/call
128					if (!%\$fields exists \$fields->{lc 'TimeTaken'});
129	74	212µs	77	71.2ms	push @data, [Type => \$t->Type] # spent 71.2ms making 74 calls to RT::Transaction::NewValue, avg 963µs/call # spent 2µs making 3 calls to DBIx::SearchBuilder::Record::DESTROY, avg 600ns/call
130					if (!%\$fields exists \$fields->{lc 'Type'});
131	74	280µs	77	44.3ms	push @data, [Field => \$t->Field] # spent 44.3ms making 73 calls to DBIx::SearchBuilder::Record::__ANON_[DBIx/SearchBuilder/Record.pm:423], avg 607µs/call # spent 24µs making 1 call to DBIx::SearchBuilder::Record::AUTOLOAD # spent 1µs making 3 calls to DBIx::SearchBuilder::Record::DESTROY, avg 467ns/call
132					if (!%\$fields exists \$fields->{lc 'Field'});
133	74	294µs	74	611ms	push @data, [OldValue => \$t->OldValue] # spent 611ms making 74 calls to RT::Transaction::Description, avg 8.26ms/call
134					if (!%\$fields exists \$fields->{lc 'OldValue'});
135	74	293µs	74	684ms	push @data, [NewValue => \$t->NewValue] # spent 684ms making 74 calls to RT::Transaction::Content, avg 9.24ms/call
136					if (!%\$fields exists \$fields->{lc 'NewValue'});
137					push @data, [Data => \$t->Data]
138					if (!%\$fields exists \$fields->{lc 'Data'});
139	74	156µs	74	49µs	push @data, [Description => \$t->Description] # spent 49µs making 74 calls to DBIx::SearchBuilder::Record::DESTROY, avg 659ns/call
140	74	496µs	74	2.00ms	if (!%\$fields exists \$fields->{lc 'Description'}); # spent 2.00ms making 74 calls to DBIx::SearchBuilder::Record::new, avg 27µs/call
141	74	370µs	148	79.5ms	push @data, [Content => \$t->Content] # spent 70.5ms making 74 calls to DBIx::SearchBuilder::Record::__ANON_[DBIx/SearchBuilder/Record.pm:423], avg 953µs/call # spent 8.96ms making 74 calls to RT::User::Load, avg 121µs/call
142	74	313µs	74	2.22ms	if (!%\$fields exists \$fields->{lc 'Content'}); # spent 2.22ms making 74 calls to DBIx::SearchBuilder::Record::__ANON_[DBIx/SearchBuilder/Record.pm:423], avg 30µs/call
143					
144	74	310µs	74	49.4ms	# spent 49.4ms making 73 calls to DBIx::SearchBuilder::Record::__ANON_[DBIx/SearchBuilder/Record.pm:423], avg 676µs/call # spent 20µs making 1 call to DBIx::SearchBuilder::Record::AUTOLOAD
145					if (!%\$fields exists \$fields->{lc 'Content'}) {
146					my \$creator = RT::User->new(\$session{currentUser});
147	74	65µs			\$creator->Load(\$t->Creator);
148	74	18µs			push @data, [Creator => \$creator->Name];
149	74	169µs	74	738µs	} # spent 738µs making 74 calls to RT::Transaction::Attachments, avg 10µs/call
150	74	151µs	74	1.05ms	push @data, [Created => \$t->Created] # spent 1.05ms making 74 calls to DBIx::SearchBuilder::Next, avg 14µs/call
151					if (!%\$fields exists \$fields->{lc 'Created'});
152					
153					if (!%\$fields exists \$fields->{lc 'Attachments'}) {
154					my \$attachlist;
155					my \$attachments = \$t->Attachments;
156					while (my \$a = \$attachments->Next) {
157					my \$size = length(\$a->Content '');
158	74	95µs			if (\$size > 1024) { \$size = int(\$size/102.4)/10 . "k" }
159					else { \$size = "b" }
160					my \$name = (defined \$a->Filename and length \$a->Filename) ? \$a->Filename : "untitled";
161	74	1.16ms			\$attachlist .= "\n" . \$a->Id . ": \$name (\$size)";
162	74	323µs			}
163	74	263µs			
164					push @data, [Attachments => \$attachlist];
165					}
166	1	300ns			
167	1	300ns			my %k = map {\$_} @data;
168	1	600ns			\$o = [map {\$_->[0]} @data];
169					\$k = \%k;
170	1	64µs	75	76.9ms	# spent 76.9ms making 75 calls to DBIx::SearchBuilder::Next, avg 1.02ms/call
171	74	42µs	74	60µs	} else { # spent 60µs making 74 calls to RT::Record::Id, avg 807ns/call
172					my (@data, \$tids);
173	74	20µs			\$format = "s";
174	74	9µs			\$format = "l" if (%\$fields);
175	74	9µs			
176					while (my \$t = \$trans->Next) {
177					my \$tid = \$t->Id;
178					
179					if (\$format eq "l") {
180					\$tids .= "," if \$tids;
181					\$tids .= \$tid;
182	1	500ns			} else {
183	1	100ns			push @\$, \$tid;
184	1	2µs			\$k->{\$tid} = \$t->Description;
185	1	100ns			}
186	1	2µs			}
187					
188					if (\$format eq "l") {
189					my @tid;
190	1	200ns			push @tid, "ticket/\$id/history/id/\$tids";
191	1	40µs	1	7.24s	my \$fieldstring; # spent 7.24s making 1 call to HTML::Mason::Request::subexec
192					foreach my \$key (keys %\$fields) {
193					\$fieldstring .= "," if \$fieldstring;
194					\$fieldstring .= \$key;
195	1	10µs			}
196					my (\$content, \$forms);
197					\$m->subexec("/REST/1.0/show",
198					id => \@tid,
199	74	62µs			format => \$format,
200	74	210µs	74	592µs	fields => \$fieldstring); # spent 592µs making 74 calls to DBIx::SearchBuilder::Count, avg 8µs/call
201	74	195µs			return [\$c, \$o, \$k, \$e];
202					}
203					}
204	74	1.16ms			
205					if (!\$c) {
206					my \$sub = \$trans->Count();
207					\$c = "# \$sub/\$total (\$args/total)";
208					}
209					
210					return [\$c, \$o, \$k, \$e];
211					
212					#line 1 "/opt/rt4/share/html/REST/1.0/Forms/ticket/history"
213					# BEGIN BPS TAGGED BLOCK {{{
214					#
215					# COPYRIGHT:
216					#
217					# This software is Copyright (c) 1996-2015 Best Practical Solutions, LLC
218					# <sales@bestpractical.com>
219					#
220					# (Except where explicitly superseded by other copyright notices)

Line	State ments	Time on line	Calls	Time in subs	Code
221					#
222					#
223					# LICENSE:
224					#
225					# This work is made available to you under the terms of Version 2 of
226					# the GNU General Public License. A copy of that license should have
227					# been provided with this software, but in any event can be snarfed
228					# from www.gnu.org.
229					#
230					# This work is distributed in the hope that it will be useful, but
231					# WITHOUT ANY WARRANTY; without even the implied warranty of
232					# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
233					# General Public License for more details.
234					#
235					# You should have received a copy of the GNU General Public License
236					# along with this program; if not, write to the Free Software
237					# Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA
238					# 02110-1301 or visit their web page on the internet at
239					# http://www.gnu.org/licenses/old-licenses/gpl-2.0.html.
240					#
241					#
242					# CONTRIBUTION SUBMISSION POLICY:
243					#
244					# (The following paragraph is not intended to limit the rights granted
245					# to you to modify and distribute this software under the terms of
246					# the GNU General Public License and is only of importance to you if
247					# you choose to contribute your changes and enhancements to the
248					# community by submitting them to Best Practical Solutions, LLC.)
249					#
250					# By intentionally submitting any modifications, corrections or
251					# derivatives to this work, or any other work intended for use with
252					# Request Tracker, to Best Practical Solutions, LLC, you confirm that
253					# you are the copyright holder for those contributions and you grant
254					# Best Practical Solutions, LLC a nonexclusive, worldwide, irrevocable,
255					# royalty-free, perpetual, license to use, copy, create derivative
256					# works based on those contributions, and sublicense and distribute
257					# those contributions and any derivatives thereof.
258					#
259					# END BPS TAGGED BLOCK }}}}
260					# REST/1.0/Forms/ticket/history
261					#
262					;return;
263);
264					'declared_args' => {
265					'\$args' => { default => ' undef' },
266					'\$fields' => { default => ' undef' },
267					'\$format' => { default => ' undef' },
268					'\$id' => { default => undef }
269					},
270					'load_time' => 1427791477,
271)
272					}
273					;