

Foreman - Bug #24384

Initial message from request does not have request/session ID

07/25/2018 01:31 PM - Lukas Zapletal

Status: New	
Priority: Normal	
Assignee:	
Category:	
Target version:	
Difficulty:	Fixed in Releases:
Triaged: No	Found in Releases: 1.18.1
Bugzilla link:	Red Hat JIRA:
Pull request:	

Description

It looks like the very first message from request does not have request and session ID set. I can only reproduce this on Foreman 1.18 and not develop, looks like something changed.

```
Wed 2018-07-25 14:23:11.655367 BST [s=9089292e04d14210b90ef9a91ee623a4;i=4017;b=05e4cce6965c4b3f85052935aa3798dc;m=15f8c5c5d;t=
  PRIORITY=6
  _SYSTEMD_SLICE=system.slice
  _BOOT_ID=05e4cce6965c4b3f85052935aa3798dc
  _MACHINE_ID=ce37bff78932646b63dbb85e691a3808
  _HOSTNAME=next.nat.lan
  _TRANSPORT=journal
  _CAP_EFFECTIVE=0
  _SYSTEMD_CGROUP=/system.slice/httpd.service
  _SYSTEMD_UNIT=httpd.service
  _UID=993
  _GID=991
  SYSLOG_IDENTIFIER=foreman
  SYSLOG_FACILITY=176
  FOREMAN_LOGGER=app
  _COMM=ruby
  _EXE=/opt/rh/rh-ruby24/root/usr/bin/ruby
  _SELINUX_CONTEXT=system_u:system_r:passenger_t:s0
  _PID=2761
  _CMDLINE=Passenger RackApp: /usr/share/foreman
  MESSAGE=Started GET "/widgets/71" for 192.168.199.1 at 2018-07-25 14:23:11 +0100
  _SOURCE_REALTIME_TIMESTAMP=1532524991655367
```

The next message in the request already have those ids set:

```
Wed 2018-07-25 14:23:11.660500 BST [s=9089292e04d14210b90ef9a91ee623a4;i=4018;b=05e4cce6965c4b3f85052935aa3798dc;m=15f8c7aa9;t=
  PRIORITY=6
  _SYSTEMD_SLICE=system.slice
  _BOOT_ID=05e4cce6965c4b3f85052935aa3798dc
  _MACHINE_ID=ce37bff78932646b63dbb85e691a3808
  _HOSTNAME=next.nat.lan
  _TRANSPORT=journal
  _CAP_EFFECTIVE=0
  _SYSTEMD_CGROUP=/system.slice/httpd.service
  _SYSTEMD_UNIT=httpd.service
  _UID=993
  _GID=991
  SYSLOG_IDENTIFIER=foreman
  SYSLOG_FACILITY=176
  FOREMAN_LOGGER=app
  _COMM=ruby
```

```
_EXE=/opt/rh/rh-ruby24/root/usr/bin/ruby
_SELINUX_CONTEXT=system_u:system_r:passenger_t:s0
_PID=2761
_CMDLINE=Passenger RackApp: /usr/share/foreman
REMOTE_IP=192.168.199.1
SESSION=9c3aca5b-23c3-4ab0-9f88-b1219acf1975
MESSAGE=Processing by DashboardController#show as HTML
REQUEST=8b96f8bd-898d-41cb-a347-dcd3d7a853e5
_SOURCE_REALTIME_TIMESTAMP=1532524991660500
```

I am trying this with Foreman 1.18.

History

#1 - 07/25/2018 02:33 PM - Lukas Zapletal

REMOTE_IP is missing as well, if I add debug line to the lib/middleware/logging_context.rb I see that Rails logs this BEFORE this middleware is called.

```
2018-07-25T15:31:02 info app Started GET "/" for 192.168.199.1 at 2018-07-25 15:31:02 +0100
2018-07-25T15:31:02 5ba9652d info app Setting request: 5ba9652d-3091-4676-9bea-f077c6596fa2
2018-07-25T15:31:02 5ba9652d info app Processing by DashboardController#index as HTML
```

#2 - 07/25/2018 02:55 PM - Lukas Zapletal

I think it was always like this, so this is not a regression.

We can improve our logging stack by supporting tagging in our logger implementation:

<https://blog.bigbinary.com/2016/06/28/rails-5-supports-logging-errors-with-tagged-logging.html>

Rails 5+ makes use of that:

```
if logger.respond_to?(:tagged)
  logger.tagged(compute_tags(request)) { call_app(request, env) }
else
```

then we would simply put request_id into list of tags to be sent along with all requests, this would replace our logging_context.rb implementation. Tags can be strings or Proc objects so it can be evaluated to anything.