

Ruby - Bug #8975

Confusing code sample for assert_send

10/02/2013 07:48 AM - agrimm (Andrew Grimm)

Status:	Closed	Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
Priority:	Normal	
Assignee:	zzak (zzak _)	
Target version:		
ruby -v:	ruby 2.0.0p247 (2013-06-27 revision 41674) [x86_64-linux]	
Description		
Test::Unit::Assertions#assert_send has the code sample		
<pre>assert_send([[1, 2], :member?, 1]) # -> pass assert_send([[1, 2], :member?, 4]) # -> fail</pre>		
Having an array within an array is somewhat confusing. When I first read it, I ignored the extra square brackets. A less confusing example would be		
<pre>assert_send(["Hello world", :include?, "Hello"]) # -> pass assert_send(["Hello world", :include?, "Goodbye"]) # -> fail</pre>		

Associated revisions

Revision 4370f84653cb15f08e188c809c3846297a877f99 - 11/09/2013 05:12 AM - zzak (zzak _)

- lib/test/unit/assertions.rb: [DOC] better example for assert_send()
Patch by Andrew Grimm [Bug #8975]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43597 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4370f846 - 11/09/2013 05:12 AM - zzak (zzak _)

- lib/test/unit/assertions.rb: [DOC] better example for assert_send()
Patch by Andrew Grimm [Bug #8975]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@43597 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 10/02/2013 09:18 PM - zzak (zzak _)

Thanks for the report Andrew!

I'm really glad we got a doc bug reported for TestUnit, this confirms my suspicion that people are still using TU for legacy apps and still need accurate documentation.

I will have to reconsider how we document TU..

#2 - 10/02/2013 09:19 PM - zzak (zzak _)

- Status changed from Open to Assigned

#3 - 10/03/2013 11:08 AM - agrimm (Andrew Grimm)

To be honest, it's not so much a "legacy app" issue as a "legacy programmer" issue.

I started programming back in Ruby 1.8.6. I don't use any opinionated web development frameworks such as Rails (I don't do web development at all, actually!), so nothing told me to switch to minitest. I used 'require "test/unit"' back then, and I still use it now. I don't actually know whether I use test/unit or whether Ruby 1.9/2.0 uses minitest with a compatibility layer. I also don't know what advantages there are in switching from test/unit to minitest.

When I switched from 1.9.x to 2.0, the way that tests were outputted changed, in that you usually don't get a series of dots being printed out for each test. That gave me the impression that whatever I was using was non-legacy.

Is test/unit considered legacy?

I think that documenting the advantages of minitest over test/unit, or that test/unit is considered legacy, may be helpful. Maybe in the documentation for test/unit?

#4 - 10/16/2013 12:13 AM - zzak (zzak _)

Please see [#8778](#) for reference.

To answer your question, lib/test (test/unit) is a legacy shim.

I don't think documenting minitest advantages will help, because minitest may not stay in the stdlib forever.

We could mention other libraries, including the full TU2 gem for those looking to upgrade. I think the natural progression is TU -> TU2, not TU -> MT

#5 - 11/09/2013 02:12 PM - zzak (zzak _)

- *Status changed from Assigned to Closed*

- *% Done changed from 0 to 100*

This issue was solved with changeset r43597.

Andrew, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

-
- lib/test/unit/assertions.rb: [DOC] better example for assert_send()
Patch by Andrew Grimm [Bug [#8975](#)]