Description

$ RAILS_ENV=test bundle exec rake db:migrate:reset

$ RAILS_ENV=test bundle exec rake test TESTOPTS="--seed 4619"

.......................................................E

Error:
RepositoryTest#test_scan_changesets_for_issue_ids:
NoMethodError: undefined method `body' for nil:NilClass

test/test_helper.rb:253:in `mail_body'
test/test_helper.rb:238:in `assert_mail_body_match'
test/unit/repository_test.rb:275:in `block in test_scan_changesets_for_issue_ids'
test/unit/repository_test.rb:271:in `each'
test/unit/repository_test.rb:271:in `test_scan_changesets_for_issue_ids'

bin/rails test test/unit/repository_test.rb:237

The test failed because Settings.plain_text_mail is not set.

Solved by the following patch.

diff --git a/test/unit/repository_test.rb b/test/unit/repository_test.rb
index 1d7d572c3..18bec60f0 100644
--- a/test/unit/repository_test.rb
+++ b/test/unit/repository_test.rb
@@ -235,49 +235,53 @@
  
 class RepositoryTest < ActiveSupport::TestCase
 end

-  
-    def test_scan_changesets_for_issue_ids
-      Setting.default_language = 'en'
-      Setting.commit_ref_keywords = 'refs , references, IssueID'
-      Setting.commit_update_keywords = [
-        {'keywords' => 'fixes , closes'},
-        'status_id' => IssueStatus.where(:is_closed => true).first.id,
-        'done_ratio' => '90'}
-      ]
-      Setting.default_language = 'en'
-      ActionMailer::Base.deliveries.clear
-
-      # make sure issue 1 is not already closed
-      fixed_issue = Issue.find(1)
-      assert !fixed_issue.closed?
-      old_status = fixed_issue.status


with_settings :notified_events => %w(issue_added issue_updated) do
  with_settings(
    :default_language => 'en',
    :commit_ref_keywords => 'refs , references, IssueID',
    :commit_update_keywords => [
      
      'keywords' => 'fixes , closes',
      'status_id' => IssueStatus.where(:is_closed => true).first.id,
      'done_ratio' => '90'
    ],
  ),
  :notified_events => %w(issue_added issue_updated),
  :plain_text_mail => '0'
) do
  ActionMailer::Base.deliveries.clear
  # make sure issue 1 is not already closed
  fixed_issue = Issue.find(1)
  assert !fixed_issue.closed?
  old_status = fixed_issue.status
  Repository.scan_changesets_for_issue_ids
  end
  assert_equal [101, 102], Issue.find(3).changeset_ids
  
  # fixed issues
  fixed_issue.reload
  assert fixed_issue.closed?
  assert_equal 90, fixed_issue.done_ratio
  assert_equal [101], fixed_issue.changeset_ids
  
  # issue change
  journal = fixed_issue.journals.reorder("created_on desc").first
  assert_equal User.find_by_login("dlopper").user,
  assert_equal 'Applied in changeset r2.', journal.notes
  
  # 5 email notifications, 2 for #1, 3 for #2
  assert_equal 5, ActionMailer::Base.deliveries.size
  ActionMailer::Base.deliveries.first(2).each do |mail|
    assert_not_nil mail
    assert mail.subject.starts_with?("[
    
    assert_mail_body_match("Status changed from #{old_status} to #{fixed_issue.status}", mail)
  end
  
  assert_equal [101, 102], Issue.find(3).changeset_ids
  
  # fixed issues
  fixed_issue.reload
  assert fixed_issue.closed?
  assert_equal 90, fixed_issue.done_ratio
  assert_equal [101], fixed_issue.changeset_ids
+ # issue change
+ journal = fixed_issue.journals.reorder('created_on desc').first
+ assert_equal User.find_by_login('dlopper'), journal.user
+ assert_equal 'Applied in changeset r2.', journal.notes
+ # 5 email notifications, 2 for #1, 3 for #2
+ assert_equal 5, ActionMailer::Base.deliveries.size
+ ActionMailer::Base.deliveries.first(2).each do |mail|
+   assert_not_nil mail
+   assert mail.subject.starts_with?("[#{fixed_issue.project.name} - #{fixed_issue.tracker.name} #{fixed_issue.id}]"")
+   assert_mail_body_match("Status changed from #{old_status} to #{fixed_issue.status}", mail)
+ end
- # ignoring commits referencing an issue of another project
- assert_equal [], Issue.find(4).changesets
+ # ignoring commits referencing an issue of another project
+ assert_equal [], Issue.find(4).changesets
+ end
+ end

def test_for_changeset_comments_strip

Associated revisions
Revision 18465 - 2019-09-13 10:51 - Go MAEDA

mail_body method in test/test_helper.rb raises an exception if the message is not multipart (#32025).

This change fixes random test failures.

History
#1 - 2019-09-13 02:47 - Go MAEDA
- Target version set to Candidate for next major release

Thank you for catching and investigating the issue.

The root cause of the issue is that mail_body method in test/test_helper.rb supports only multi-part messages and does not support plaintext messages.

I think it is better to change mail_body method to support plaintext messages as well.

```diff
(diff --git a/test/test_helper.rb b/test/test_helper.rb)
index 390ea92b4..6a6fd7fa5 100644
--- a/test/test_helper.rb
+++ b/test/test_helper.rb
@@ -250,7 +250,7 @@ class ActiveSupport::TestCase
end
```
def mail_body(mail)
  - mail.parts.first.body.encoded
  + (mail.multipart? ? mail.parts.first : mail).body.encoded
end

# Returns the lft value for a new root issue

#2 - 2019-09-13 03:58 - Yuichi HARADA

Go MAEDA wrote:

```
The root cause of the issue is that mail_body method in test/test_helper.rb supports only multi-part messages and does not support plaintext messages.

I think it is better to change mail_body method to support plaintext messages as well.

[...]
```

+1
I agree. Setting of Setting/plain_text_mail is not directly related to the acquisition of mail_body.

#3 - 2019-09-13 10:52 - Go MAEDA

- Subject changed from RepositoryTest#test_scan_changesets_for_issue_ids randomly fails to mail_body method in test/test_helper.rb raises an exception if the message is not multipart
- Status changed from New to Closed
- Assignee set to Go MAEDA
- Target version changed from Candidate for next major release to 4.1.0

Committed the fix. Thank you for catching the issue.