Wrong use of refute_includes in tests

The `refute_includes` assertion expects the collection first and the object to test for after. The test in `source:/trunk/test/functional/queries_controller_test.rb@17558#L627` uses the object then collection form though.

It seems most of Redmine tests use `source:/trunk/test/test_helper.rb@17558#L205` `assert_not_include` which expects the object then collection form. The following patch corrects this problem:

```ruby
--- a/test/functional/queries_controller_test.rb
+++ b/test/functional/queries_controller_test.rb
@@ -645,7 +645,7 @@
     assert_includes ['OnlineStore - Systemwide visible version', "7", "open"], json
 # response doesn't include non visible version
- refute_includes ['Private child of eCookbook - Private Version of public subproject', "6", "open"], json
+ assert_not_includes ['Private child of eCookbook - Private Version of public subproject', "6", "open"], json
     end

def test_subproject_filter_time_entries_with_project_id_should_return_filter_values
```

Associated revisions

Revision 17559 - 2018-10-02 04:12 - Go MAEDA

Wrong use of refute_includes in tests (#29708).

Patch by Felix Schäfer.

History

#1 - 2018-10-02 02:10 - Mizuki ISHIKAWA

That makes sense to me.

```
refute_includes json, ['Private child of eCookbook - Private Version of public subproject", "6", "open"]
```

Although the test works with the above code, refute_includes is only used in redmine at three places.

assert_not_includes is more familiar.

#2 - 2018-10-02 03:38 - Go MAEDA

- Category set to Code cleanup/refactoring
- Target version set to 4.1.0
Committed. Thanks.