Redmine - Defect #5793

Use utf8 character sets and collations in mysql

2010-07-01 13:41 - Aleksandar Pavic

Status:	Closed	**Start date:**	2010-07-01
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Database	**Estimated time:**	0.00 hour
Target version:			
Resolution:	Invalid	**Affected version:**	0.9.4

Description

Well I don't know is this really a defect, but I had to change collations to utf8_general_ci, and character set to utf8 to use latin characters.

After that everything worked as charm.

History

#1 - 2010-07-01 14:08 - Felix Schäfer

- Status changed from New to Closed

- Resolution set to Invalid

Redmine uses UTF-8 out of the box, MySQL doesn't create tables in UTF-8 out of the box but that is also described in the MySQL-specific part of the installation.

#2 - 2010-10-14 11:21 - Jérémy Lal

- Status changed from Closed to Reopened

It could be improved, though :
doing a pristine install, when database.yml contains encoding: utf8
all data recorded to mysql is properly encoded,
EXCEPT for the data that is loaded by redmine:load_default_data task.

#3 - 2010-10-14 11:22 - Jérémy Lal

i meant when default database encoding is Latin, rails encoding utf8
makes sure table are utf8, except for the ones that corresponds to the data
loaded by load_default_data task.

#4 - 2010-10-14 19:22 - Felix Schäfer

Jérémy Lal wrote:

> i meant when default database encoding is Latin, rails encoding utf8
> makes sure table are utf8, except for the ones that corresponds to the data
> loaded by load_default_data task.

Rails doesn't do nothing, saying encoding: utf8 in the database.yml just makes sure rails sends the data as UTF-8 to the DB engine. If the DB engine is configured with latin1 as collation, rails can't do anything against it.

Could you please describe what exactly you have done to get that error on a pristine install?

#5 - 2010-10-14 23:38 - Jérémy Lal

Hi,
this is probably not a bug, sorry.
Still, rake db:create or db:migrate could use
CREATE TABLE roles (...) DEFAULT CHARACTER SET utf8;
and that would solve the problem of mysql db created in latin1 (by default on most mysql installs).

#6 - 2010-10-15 00:45 - Jérémy Lal

The trouble is postgres, sqlite has been defaulting to UTF8 for quite a long time,

while mysql has still some weird encoding as its default. So most postgres or sqlite
users won't be bothered, while many mysql users will.
The following patch (added to config/initializers/) could help,
by creating tables with default charset utf8, regardless of the character_set_database.

```
require 'active_record'
require 'active_record/connection_adapters/abstract_adapter'
module ActiveRecord
  module ConnectionAdapters
    class MysqlAdapter < AbstractAdapter
      def create_table(table_name, options = {}) #:nodoc:
        encoding = @config[:encoding]
        if encoding
          options = options.reverse_merge(:options => "DEFAULT CHARSET=#{encoding}")
        end
        super(table_name, options.reverse_merge(:options => "ENGINE=InnoDB"))
      end
    end
  end
end
```

The patch consists in adding the three lines [if ... end].

Feel free to close the bug as invalid,
Jérémy.

#7 - 2010-10-17 17:49 - Felix Schäfer

- Status changed from Reopened to Closed

Jérémy Lal wrote:

> Still, rake db:create or db:migrate could use
> CREATE TABLE roles (...) DEFAULT CHARACTER SET utf8;
> and that would solve the problem of mysql db created in latin1 (by default on most mysql installs).

Then it's something you should report and discuss with the rails team and/or the maintainer of the mysql gem :-) The docs make it clear you should use mysql with UTF-8 as the collation, I don't think we need (yet another…) rails core patch for that.

#8 - 2010-10-18 14:15 - Jérémy Lal

Agreed, done here :
http://rails.lighthouseapp.com/projects/8994/tickets/5830-mysql-adapter-should-create-table-with-default-charset

Regards,
Jérémy.