

Redmine - Feature #1341

keep consistency between browser encoding and mysql database encoding

2008-06-01 13:24 - Gilles Ballanger

Status:	Closed	Start date:	2008-06-01
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Resolution:	Invalid		

Description

Hello,

after trying to lazily import issue directly in mysql database (I know it's very bad to do like this, it's better using ruby importation script via redmine API) I see issue subject (and description too) badly utf-8 encoded :

if I import record via SQL using

```
INSERT INTO `issues` (`tracker_id`, `project_id`, `subject`, `description`, `due_date`, `category_id`, `status_id`, `assigned_to_id`, `priority_id`, `fixed_version_id`, `author_id`, `lock_version`, `created_on`, `updated_on`, `start_date`, `done_ratio`, `estimated_hours`) VALUES (4, 1, 'é', 'é', NULL, NULL, 1, NULL, 4, 5, 3, 0, '2008-05-30 14:19:43', '2008-05-30 14:19:43', '2008-05-30', 0, NULL);
```

the resulting database dump for this record is

```
INSERT INTO `issues` (`id`, `tracker_id`, `project_id`, `subject`, `description`, `due_date`, `category_id`, `status_id`, `assigned_to_id`, `priority_id`, `fixed_version_id`, `author_id`, `lock_version`, `created_on`, `updated_on`, `start_date`, `done_ratio`, `estimated_hours`) VALUES (234, 4, 1, 0xc3a9, 0xc3a9, NULL, NULL, 1, NULL, 4, 5, 3, 0, '2008-05-30 14:19:43', '2008-05-30 14:19:43', '2008-05-30', 0, NULL);
```

and the result on browser show a '❖' char in place of 'é'

If I insert an issue via browser with subject='é' and description='é' the dumped database is

```
INSERT INTO `issues` (`id`, `tracker_id`, `project_id`, `subject`, `description`, `due_date`, `category_id`, `status_id`, `assigned_to_id`, `priority_id`, `fixed_version_id`, `author_id`, `lock_version`, `created_on`, `updated_on`, `start_date`, `done_ratio`, `estimated_hours`) VALUES (235, 1, 1, 0xc383c2a9, 0xc383c2a9, NULL, NULL, 1, NULL, 4, NULL, 3, 0, '2008-06-01 13:14:20', '2008-06-01 13:14:20', '2008-06-01', 0, NULL);
```

=> the 'é' char was coded in hex c3 83 c2 a9 (the correct encoding is c3 a9)

This produce "Ã©" in place of "é" in mysql database dump but a correct é char in issue

My knowledge in ruby are not sufficient to reproduce this kind of string encoding interpretation but I do it in python :

first I encode 'é' char in utf-8 by:

```
>>> unicode("é","utf-8").encode("utf-8")
'\xc3\xa9'
```

If i take each value, declare it as unicode string and recode it in utf-8, I have the same bad coding behavior

```
>>> (u"\xc3").encode("utf-8")
'\xc3\x83'
>>> (u"\xa9").encode("utf-8")
```

'\xc2\xa9'

so perhaps there is a double encoding conversion somewhere between what is send from browser to what is write in database ?

Once again importing directly in database is a very bad idea (this is a perfect example) but meanwhile this inconsistency between database coding and page rendering can be source of problem in future ...

History

#1 - 2008-06-01 20:26 - Thomas Löber

What are the values of your MySQL variables?

```
mysql> show variables like 'character%';
```

Variable_name	Value
character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

You may set the MySQL character set variables in my.cnf (e.g. /etc/mysql/my.cnf).

For the server:

```
[mysqld]
character-set-server = utf8
```

For the client:

```
[client]
default-character-set = utf8
```

The character set setting for the Rails connection to MySQL is in config/database.yml:

```
production:
  adapter: mysql
  ...
  encoding: utf8
```

#2 - 2008-06-02 15:43 - Gilles Ballanger

- Status changed from New to Resolved

Original situation :

```
mysql> show variables like 'character%';
```

Variable_name	Value
character_set_client	latin1
character_set_connection	latin1
character_set_database	latin1
character_set_filesystem	binary
character_set_results	latin1
character_set_server	latin1
character_set_system	utf8
character_sets_dir	/usr/share/mysql/charsets/

all wrong :(...

after adapting server client and redmine configuration files consistency is back. :)

of course the issues already in database with bad encoding appear with wrong character set but new one with "good" utf-8 encoding are correctly display.

Thanks for your solution.

#3 - 2008-06-02 19:00 - Jean-Philippe Lang

- *Status changed from Resolved to Closed*

- *Resolution set to Invalid*