Vbulletin3.8.7 to Discourse Guide

1. Convert vBulletin(MySQL) database to utf8

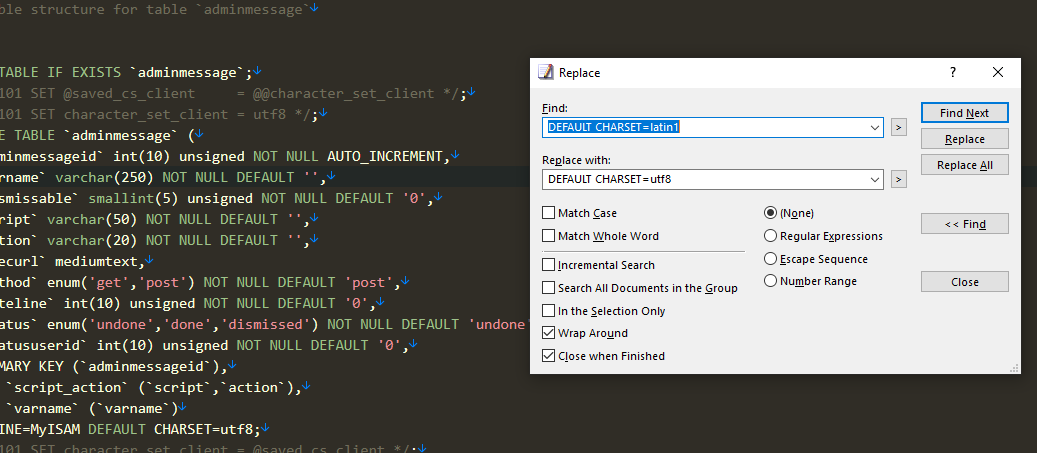
Discourse(works on ruby on rails) uses postgresql and it’s default charset is utf8.

But our vBulletin database charset is latin1 and tables use windows-1256(Arabic) charset.

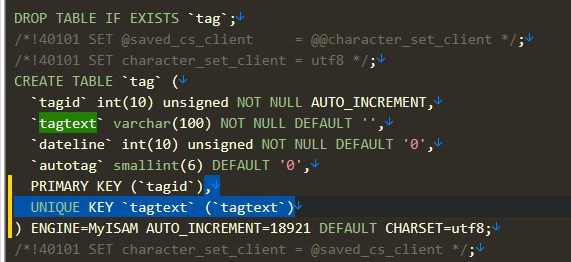
I did this work on windows using sqlyog.

1. Convert table charset from latin1 to utf8

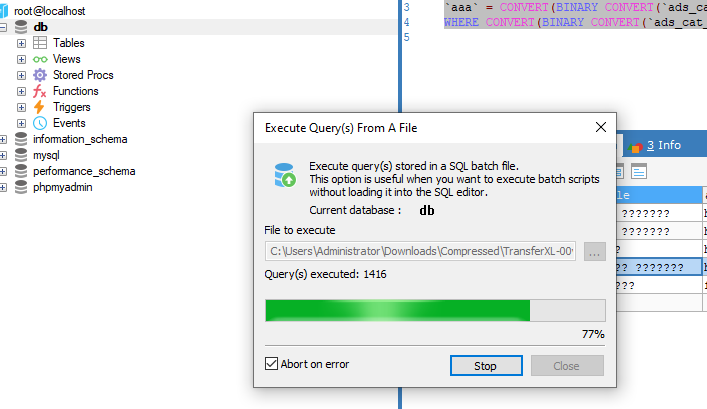
**Open db.sql(from vBulletin) with editor(emeditor, notepad++, don’t open with notepad.exe) and replace “DEFAULT CHARSET=latin1” string to “DEFAULT CHARSET=utf8”**



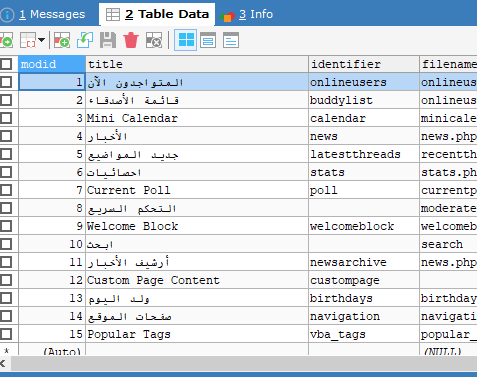
Remove **UNIQUE KEY `tagtext` (`tagtext`)** from tag table.



**Import this db by sqlyog**



**I convert table field using my conv.php.**



We can get clear Arabic text here.

There are many arabic text tables here but we can do it using conv.php.

You can modify conv.php according to table type.

This work can take 1 ~ 2hours.

1. Database prepare and upload .

Next we have to export this table and upload it to vps using scp or ftp tool. I used winscp.

1. Install Discourse on ubuntu 22.04 for development

We assume that server is very clean state.

1. Install Mysql

First of all we have to install mysql. Run terminal and input this command.

**sudo apt-get install libmysqlclient-dev mysql-server**

After finishing installing MySQL check its status:

**sudo service mysql status**

If MySQL service is not running/active:

**sudo service mysql start**

1. Install dependencies

Run this command in terminal.

You can work on /root(~) directory.

**wget -qO-** [**https://raw.githubusercontent.com/discourse/install-rails/master/linux**](https://raw.githubusercontent.com/discourse/install-rails/master/linux)

After that you can see file linux.

Please input **nano linux.**

Then you can show file content like follows;

Run this file like **bash < linux**

There are no libwebp6 now. So error happens and we have to modify **install-imagemagick** file.

Open it with nano or vi and modify libwebp6 to libwebp7.

And run this command again.

**sudo ./install-imagemagick**

After install it we have to erase above part that performed before.

Open **linux** file and erase these part.

log\_info "Installing Git ..."

sudo apt-get -y install git

log\_info "Installing build essentials ..."

sudo apt-get -y install build-essential

log\_info "Installing libraries for common gem dependencies ..."

sudo apt-get -y install libxslt1-dev libcurl4-openssl-dev libksba8 libksba-dev libreadline-dev libssl-dev zlib1g-dev libsnappy-dev

log\_info "Installing sqlite3 ..."

sudo apt-get -y install libsqlite3-dev sqlite3

log\_info "Installing Postgres ..."

sudo apt-get -y install postgresql postgresql-server-dev-all postgresql-contrib libpq-dev

log\_info "Installing curl ..."

sudo apt-get -y install curl

log\_info "Installing Redis ..."

curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/redis.list

sudo apt-get update

sudo apt-get -y install redis

log\_info "Installing ImageMagick ..."

sudo apt-get -y install libtool

wget https://raw.githubusercontent.com/discourse/discourse\_docker/master/image/base/install-imagemagick

chmod +x install-imagemagick

sudo ./install-imagemagick

run **bash < linux** again. Everything will goes well.

Run these commands step by step in terminal.

We are on /root(~) directory now.

**sudo apt install build-essential checkinstall zlib1g-dev**

**git clone https://github.com/asdf-vm/asdf.git ~/.asdf --branch v0.10.0**

**git clone https://github.com/discourse/discourse.git ~/discourse**

**sudo -u postgres createuser -s "$USER"**

**wget https://www.openssl.org/source/openssl-1.1.1q.tar.gz**

**tar -zxf openssl-1.1.1q.tar.gz && cd openssl-1.1.1q**

**./config --prefix=/opt/openssl-1.1.1q --openssldir=/opt/openssl-1.1.1q shared zlib**

**make**

**make test**

**make install**

**sudo rm -rf /opt/openssl-1.1.1q/certs**

**sudo ln -s /etc/ssl/certs /opt/openssl-1.1.1q**

Since Ubuntu 20.04, system use openssl 3.0 and we can not used it. We have to use openssl 1.1.\* series.

Open ~/.bashrc file and add three lines.

**export RUBY\_CONFIGURE\_OPTS="--with-openssl-dir=/opt/openssl-1.1.1q/"**

**. $HOME/.asdf/asdf.sh**

**. $HOME/.asdf/completions/asdf.bash**

You can see

**export PATH="$HOME/.rbenv/bin:$PATH"**

**eval "$(rbenv init -)"**

at end of bashrc file.

After that Run these cmd in terminal step by step.

**source ~/.bashrc**

**asdf plugin add ruby**

**asdf install ruby 2.7.6**

**cd ~/discourse**

**bundle install**

**yarn install**

**bundle exec rake db:create**

**bundle exec rake db:migrate**

**RAILS\_ENV=test bundle exec rake db:create db:migrate**

After that please do as follows in terminal.

**cd ~/discourse**

**echo "gem 'mysql2', require: false" >> Gemfile**

**echo "gem 'php\_serialize', require: false" >> Gemfile**

**bundle install**

**sudo apt-get update**

**sudo apt-get install jhead libjpeg-turbo-progs jpegoptim gifsicle optipng**

**npm install -g svgo**

**mysql -uroot -p -e 'CREATE DATABASE vb3'**

**mysql -uroot -p vb4 < vb3.sql(this is sql file you upload.)**

you have to set root password name.

please do these commands in terminal.

**export DB\_NAME="vb4" # Change this to the name of VB4 database.**

**export DB\_USER="root"**

**export DB\_PW="" # Add the password of root user that you provided in the installation.**

**export TABLE\_PREFIX="vb\_" # Change this to correspond to your database prefix (or empty if there is not).**

**export ATTACHMENT\_DIR='/path/to/your/attachment/folder' # The path for attachments you copied from the old vb4 server.**

**export TIMEZONE="America/Los\_Angeles" # Change this if needed**

**cd ~/discourse**

**bundle exec ruby script/import\_scripts/vbulletin.rb**

1. Fixing errors and deploy

Many errors in vBulletin.rb.1

1. **Line 991 :**

def parse\_timestamp(timestamp)

Time.zone.at(@tz.utc\_to\_local(timestamp))

End

**Change as** :

def parse\_timestamp(timestamp)

Time.zone.at(timestamp)

End

1. **Line 971:**

user.suspended\_at = parse\_timestamp(user["bandate"]) to

user.suspended\_at = parse\_timestamp(user["bandate"].to\_i)

1. **Line 636:**

a.contentid to a.postid

1. **Line 575:**

attachments = mysql\_query(<<-SQL

SELECT a.attachmentid, a.contentid as postid, p.threadid

FROM #{TABLE\_PREFIX}attachment a, #{TABLE\_PREFIX}post p

WHERE a.contentid = p.postid

AND contenttypeid = 1 AND state = 'visible'

SQL

)

**To :**

attachments = mysql\_query(<<-SQL

SELECT a.attachmentid, a.postid as postid, p.threadid

FROM #{TABLE\_PREFIX}attachment a, #{TABLE\_PREFIX}post p

WHERE a.postid = p.postid

AND p.visible = '1'

SQL

)

1. **Line 413:**

sql = "SELECT a.attachmentid attachment\_id, a.userid user\_id, a.filedataid file\_id, a.filename filename,

LENGTH(fd.filedata) AS dbsize, filedata, a.caption caption

FROM #{TABLE\_PREFIX}attachment a

LEFT JOIN #{TABLE\_PREFIX}filedata fd ON fd.filedataid = a.filedataid

WHERE a.attachmentid = #{attachment\_id}"

To :

sql = "SELECT a.attachmentid attachment\_id, a.userid user\_id, a. attachmentid file\_id, a.filename filename,

filesize AS dbsize, filedata, a.filename caption

FROM #{TABLE\_PREFIX}attachment a

WHERE a.attachmentid = #{attachment\_id}"

1. **Line 134:**

SELECT userid

, username

, homepage

, usertitle

, usergroupid

, joindate

, email

, password

, salt

To :

SELECT userid

, username

, homepage

, usertitle

, usergroupid

, joindate

, lastvisit

, email

, password

, salt

1. **/root/discourse/config/application.rb :**

Please add <require ‘csv’>

**bundle exec ruby script/import\_scripts/vbulletin.rb** can take some time (5~7 hours).

After that run this commands in terminal;

**cd ~/discourse/script**

**./discource backup**

Once error happens, we have to comment raise error part in callback.rb and run again.

After that you can find .gz file under the public/backup/default directory.

Install discourse according to

**https://github.com/discourse/discourse/blob/main/docs/INSTALL-cloud.md**

Upload our backup zip file by following this url:

[**https://meta.discourse.org/t/move-your-discourse-instance-to-a-different-server/15721**](https://meta.discourse.org/t/move-your-discourse-instance-to-a-different-server/15721)

**References :**

[**https://meta.discourse.org/t/migrate-a-vbulletin-4-forum-to-discourse/54881**](https://meta.discourse.org/t/migrate-a-vbulletin-4-forum-to-discourse/54881)

[**https://meta.discourse.org/t/install-discourse-on-ubuntu-for-development/14727**](https://meta.discourse.org/t/install-discourse-on-ubuntu-for-development/14727)

[**https://meta.discourse.org/t/notes-about-installing-discourse-on-ubuntu-20-04/226363/2**](https://meta.discourse.org/t/notes-about-installing-discourse-on-ubuntu-20-04/226363/2)

[**https://meta.discourse.org/t/move-your-discourse-instance-to-a-different-server/15721**](https://meta.discourse.org/t/move-your-discourse-instance-to-a-different-server/15721)

[**https://github.com/discourse/discourse/blob/master/docs/INSTALL-cloud.md**](https://github.com/discourse/discourse/blob/master/docs/INSTALL-cloud.md)

**https://github.com/discourse/install-rails/blob/main/linux**