The lack of utf8mb4 support in Tiki before Tiki19 will cause issues if you attempt to use these special characters (This usually happens for emojis).

You can type them in but they are not properly handled after. Issues can go from characters being ignored or cause blank pages when viewing.

utf8mb4 is supported starting in Tiki19, including upgrading from previous versions. So if you upgrade

- from Tiki18 or older
- to Tiki19 or more recent (ex. you can upgrade from Tiki15 to Tiki21 in one step)

the migration script will convert your database tables.

This is a transition other PHP/MySQL-MariaDB projects also had to go through:

- https://core.trac.wordpress.org/ticket/21212
- https://www.drupal.org/node/1314214

Please note that Tiki18 and older can be installed on a utf8mb4 database, but Tiki code won't take advantage of it, and thus can't properly deal with emojis and other special characters that are in utf8mb4 but not in utf8.

Related wishes

- change the table definitions to support the new utf8mb4 character
- 4-byte string (most Emoji) Causes Wiki Cutoffs
- Changing (modernizing) Tiki smileys (we should support Emoji)
- Replace Emoticons with Emoji

Legacy discussions

http://tonyshowoff.com/articles/better-unicode-support-for-mysql-including-emoji/

We could simply convert the utf8 encoding to utf8mb4 like was done with utf8 in Tiki, but there is a slight performance penalty when using utf8mb4 over utf8.

One must also take into consideration that information stored in MySQL is stored in multibyte variable format, (when varchar is used) but not when char is, so chars (currently taking 3 bytes) will need 4 under utf8mb4. Using ascii encoding only requires one byte. It might be a good idea if we start choosing encoding in Tiki based on the need for basic text (ascii) multi language (utf8) and emoji support (utf8mb4), that would ensure the smallest and fastest database footprint. -drsassafras

In preparation for using this encoding, and also for good general database practices, we should cut down on
the log indexes that occasionally occur. These were likely introduced by mistake by coders who didn't understand the optimal size of MySQL indexes anyhow. Index are not stored in variable encoding, so the addition of an extra byte per character in converted utf8mb4 encoding could (and does in a few places) exceed MySQL maximum index size. -drssaafras

Another barrier to implementation is that utf8mb4 is supported from MySQL 5.5.3 (early 2010). Right now our current minimum MySQL version is 5.0. The easiest way to deal with this is likely to bump the minimum version number to 5.5.3. Almost all MySQL servers are at least this version at this point, I'm relatively sure. Might need to change the MySQL version checks etc. -drssaafras

drssaafras: Post-LTS versions are a great time to bump requirements. Just after Tiki 18 LTS is released, we can do so, and everyone who can't yet follow can stay on Tiki18 LTS for 5 years. See Versions. I think we should move to PHP7.1 or 7.2 as well.