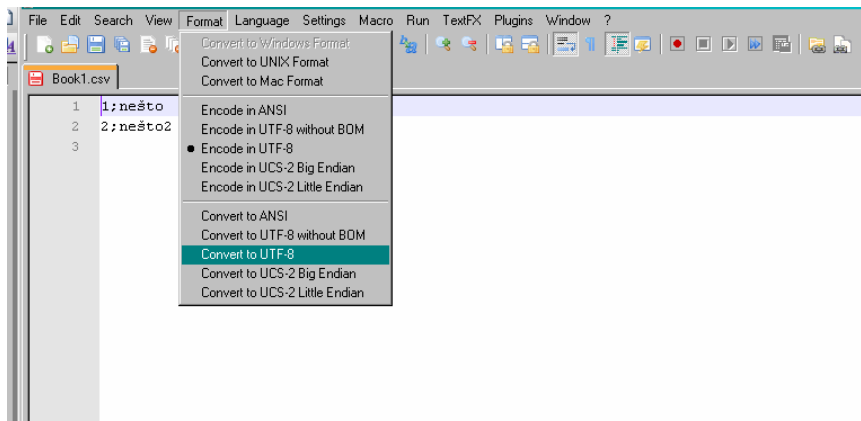


Importovanje Unicode CSV fajla u MySQL i njegovo prikazivanje u okviru PHP aplikacije

Unicode UTF-8 CSV (comma-separated values) tekst fajla, koji ste generisali ili eksportovali iz neke aplikacije kao što je npr. Excel, možemo importovati vrlo lako u MySQL via LOAD DATA INFILE komande. To važi ukoliko u tom fajlu nemamo specijalnih znakova kao što je npr. naša azbuka ili abeceda, bilo ćirilica ili latinica.

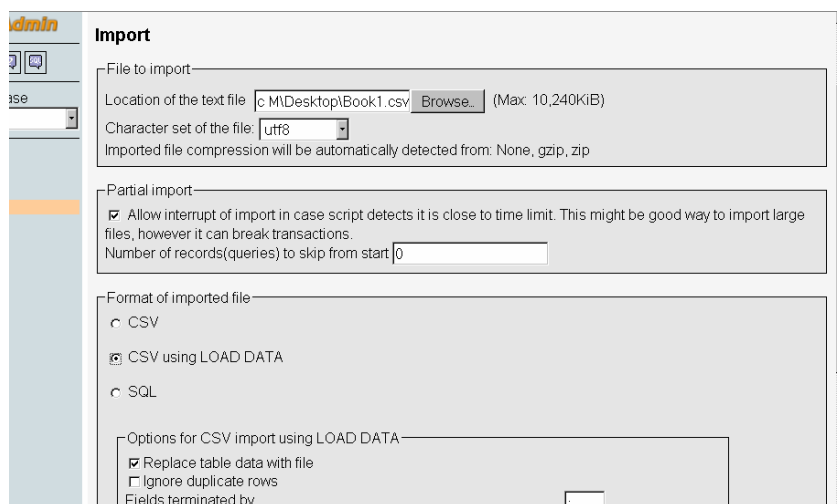
Ali ukoliko ih imamo šta i kako uraditi?

CSV podaci koji su pisani (srpskim, crnogorskim, bosanskim, hrvatskim i drugim manje sličnim jezicima) moraju se prvo i pre svega konvertovati u UTF-8 upotrebom nekog Unicode Convertera ili jednostavno učitavanjem tog CSV fajla u npr. Notepad++ gde se convertuje u UTF-8 i sačuva se kao takav.



Pre importovanja treba da proverite da li je baza i ta odredjena tabela u MySQL takodje UTF-8.

Čekirajte CSV using LOAD DATA, i ukoliko niste ništa drugo menjali u samom CSV fajlu, kliknite na GO. Ukoliko sve prodje kako treba dobićete poruku o broju uspešno importovanih redova (recorda).



Ovo je sve što se tiče importovanja.

Naravno, ako importujete podatke koji su potrebni nekoj Joomla!noj komponenti, tu se vaš posao završava, ostalo radi Joomla!.

A šta ako sami rešite da pišete neku npr. PHP aplikaciju...

Druga priča je kako te podatke iz tabele pravilno prikazati u nekoj PHP skripti.
Na šta treba obratiti pažnju?

Ono što je važno, to je da ukoliko sami radite aplikaciju, ona cela mora koristiti UTF-8 enkoding. Jedino tako možete biti sigurni da će se naša slova (ili neki drugi specijalni karakteri) pravilno prikazivati. U nekim slučajevima gde se koriste određene "string funkcije" kao što je "substr" koje koriste početnu poziciju ili dužinu stringa, potrebno je koristiti multi-byte safe function: mb_substr.

PHP komponente

Pre svega svaki element aplikacije ima svoj enkoding:

- PHP fajlovi
- PHP internal
- Izlaz ka klijentu
- Klijent browser
- Ulaz od klijenta (get, post, cookies, files)
- MySQL storage
- MySQL link
- ilineke druge ulazno-izlazne komponente

PHP fajlovi

Obavezno prilikom pisanja skripti koristite tekst editore koji u sebi sadrže mogućnost izbora enkodinga u kojem hoćete da sačuvate vaše fajlove, a to je u našem slučaju UTF-8. Uvek proverite da li ste tako i zapisali vaš PHP fajl.

Besplatni editor teksta koji to može da uradi je i već spomenuti Notepad++, koji možete skinuti sa sledećeg linka: <http://notepad-plus.sourceforge.net/uk/download.php>

PHP internal

PHP internal enkoding treba da bude u istom enkodingu kao i php fajlovi.

Za to na samom početku vaše skripte treba da stoji sledeće:

```
mb_internal_encoding('UTF-8');
```

Izlaz za klijenta

Možete podesiti i izlazni enkoding sa mb_http_output ali podrazumevano je 'pass', što znači da PHP neće dirati enkoding izlaza i on je isti kao što je i u osnovnom fajlu.

U praksi ova funkcija vam i nije potrebna, ukoliko vaše PHP fajlove zapisujete u istom enkodingu i ukoliko su svi ulazi i izlazi takođe u istom enkodingu.

Klijent browser

Da biste bili sigurni da klijent ima dobro podešen enkoding, kada prikazuje generisani HTML fajl, morate poslati sledeći "header" do klijenta:

Zapamtite: ovu PHP funkciju morate postaviti pre bilo kog drugog izlaza ka klijentu.

```
header('Content-Type: text/html; charset=UTF-8');
```

ili

U HTML headeru dodajte sledeću liniju:

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

Ulaz od strane klijenta

Postoje različiti načini na koje klijent može da šalje podatke u vašu PHP aplikaciju, kao što su GET, POST, COOKIES ili FILES. Svaki od ovih ulaza može da ima drugačiji enkoding – teorijski i treba ih proveriti. U praksi obično browser odgovara istim enkodingom kojim je i prikazana strana.

MySQL baza

Budite sigurni da ste collation vaše baze, a posebno tekstualna polja podesili da budu UTF-8. Najuobičajenije je da to bude MySQL-ov utf8_general_ci.

MySQL link

Konačno, način na koji vaša aplikacija poziva bazu menjate na sledeći način, odmah nakon otvaranja konekcije sa bazom:

```
mysql_query("SET NAMES 'utf8';", $link);  
mysql_query("SET CHARACTER SET 'utf8';", $link);
```

:)

To bi u najkraćem bilo sve!

O svemu ovde spomenutom mnogo više možete naći uz pomoć Googlea, ukoliko vam je potrebno.

Za početak ovo ovde napisano je sasvim dovoljno da uspešno u vašoj aplikaciji prikazete sva naša slova.