

# Uživatel:Josmart/Rogo/Synchronizace

## Automatizovaná synchronizace předmětů a studentů

Na serveru poběží několik CRON úloh, které by měly být spouštěny v následujícím pořadí. Je nutné zajistit, aby v době spuštění scriptu byly předchozí scripty hotovy. K tomu může pomoci soubor **status.txt**, který informuje, zda předchozí script proběhl úspěšně.

### cache\_modules.php

- vytvoří seznam předmětů **modules.csv**, určený pro ruční import do Rogo
  - UTF-8 (bez BOM)
  - delimiter = ","
  - čárky nahrazeny

```
&#44;
```

- Soubor bude využit scriptem **update\_modules.php**, který soubory naimportuje do Rogo
- Doba běhu v řádech sekund

### cache\_enrolments.php

- vytvoří tolik XML souborů, kolik je předmětů v **modules.csv**.
- Každé XML obsahuje informace o předmětu a o studentech, kteří si předmět zapsali
- Tyto XML soubory využije následně **users\_from\_SMS.php**, což je součást Rogo, k aktualizaci přiřazení studentů k předmětům.
- Doba běhu 80-90 minut, pokud zpracováváme jen data 1. LF

### update\_modules.php

- Rogo nenabízí službu, script pro automatizovaný import předmětů.
- Obsahuje i funkci, která deaktivuje předměty nenacházející se v CSV.
- Bylo možné vynechat script **cache\_modules.php** a aktualizaci modulů udělat rovnou zde. Nakešovaný soubor má ovšem tu výhodu, že jej lze do Rogo naimportovat i ručně, kdyby nastaly jakékoliv problémy.

### users\_from\_SMS.php

- každý předmět má v databázi uvedenu URL, kde se má nacházet XML s informacemi o zapsaných studentech
- Tyto XML soubory jsme předem nakešovali
- Pokud soubor obsahuje studenta, který v Rogo neexistuje, bude automaticky vytvořen.
- Pokud soubor neobsahuje studenta, který je předmětu přiřazen v databázi, bude student z předmětu vyřazen.
- V **config/config.inc.php** musí být zadáno **\$cfg\_sms\_api = 'cuni'**;
- Pro import je použita třída **plugins/SMS/cuni.class.php**, která není součástí distribuce Rogo

## LDAP autentifikace

- Přihlašování přes LDAP funguje, pokud je uživatel v Rogu již zaveden a jeho username je stejný jako v CAS. To je zajištěno pravidelnou synchronizací.
- Pokud zadáváme uživatele ručně a použijeme funkci **Získat detaily z LDAP**, automaticky se do formuláře vyplní údaje nalezené osoby. Username bude číslo osoby UK.
- Přihlásí-li se uživatel úspěšně přes LDAP, je hash jeho hesla uložen do DB a pro příští přihlášení již není potřeba kontaktovat LDAP. Nejdříve se zkouší heslo v interní databázi.
- Je nastavena expirace hesla 365 dní. Pokud se uživatel do této doby od založení účtu nepřihlásí, heslo expiruje. Takových případů je hodně. Uživatele totiž zakládáme automaticky, i když to nepotřebují. Proto je kvůli zamezení problémům s přihlašování vhodně jednou ročně (na začátku semestru) dotazem v SQL nastavit expiraci do dostatečně vzdálené budoucnosti.

By default the expiry date is 30 days. This setting is in the config file as `$cfg_password_expire` so you could increase that. This would take affect on all new users and any current user who has a password reset. Setting `password_expire` to NULL will cause the password never to expire which of course has its own risks.