

Úvod

Tato dokumentace pojednává o řešení prvního projektu do předmětu IPP. Jedná se o skript v jazyce PHP ve verzi 5.3.3. Účelem skriptu je vyhodnocení dotazu nad vstupním XML souborem. Dotaz je podobný příkazu `SELECT` jazyka SQL.

V dokumentaci se nachází způsob řešení zpracování parametrů, ověření a aplikace dotazu.

Zpracování parametrů

K zpracování parametrů jsem využil standardní funkci jazyka PHP, `getopt()`. Tato funkce dokáže zpracovat zkrácené i dlouhé parametry programu. Tyto parametry se následně uloží do asociativního pole, kde klíčem je zadaný parametr a hodnotou prvku je zadaná hodnota parametru.

Při kontrole parametrů také kontroluji, či některý z parametrů nebyl zadán vícekrát. Kontrolu provádím pomocí funkce `array_count_values()`, když se některá hodnota nachází v poli vícekrát, tak byli zadané duplicitní argumenty.

Problémem také bylo načítání krátkých parametrů, protože funkce `getopt()` dokáže kombinovat krátké parametry. Proto například `-nabc` vyhodnotí, že byl zadán argument `-n`. Vyřešil jsem to tak, že procházím všechny parametry s jednou pomlčkou v poli `argv[]` a kontroluji, zda li je řetězec rovný řetězci `"-n"`. Načtené parametry následně vrátím v poli `$options`.

Načtení vstupního XML

Na základě parametru `--input` načítám do proměnné celý obsah zadaného souboru. Zadanou cestu provádím na absolutní cestu pomocí funkce `realpath()`. Když funkcí `is_link()` zjistím, že cesta k souboru je zadaná symbolickým odkazem, tak využívám funkci `readlink()`, která mi zjistí skutečnou cestu k souboru.

Poté použiji funkci `simplexml_load_string()`, které předám proměnnou, která obsahuje obsah vstupního souboru. Táto funkce vytvoří objekt typu `SimpleXMLElement`, díky čemu poté můžu jednoduše pracovat se zadaným XML souborem.

Ověření a rozdělení dotazu

`SELECT` dotaz je zadaný přepínačem `--query` jako textový řetězec, nebo přepínačem `--qf` jako soubor obsahující dotaz.

Dotaz se skládá z částí `SELECT`, `FROM`, `LIMIT` a `WHERE`. Přičemž `LIMIT` a `WHERE` nemusí být zadané vůbec. Na ověření, zda je dotaz zadaný správně využívám regulární výraz, který si zároveň zapamatuje také element části `SELECT` a číselnou hodnotu části `LIMIT`. Tyto hodnoty se uloží do pole s klíčem podle jména části dotazu. Stejným regulárním výrazem si také zapamatuji a uložím všechno co se nachází za klíčovým slovem `FROM`. Nejprve zjistím, zda se za klíčovým slovem `FROM` nachází element, poté hledám část `WHERE`. Když je klíčové slovo `WHERE` zadané, tak si uložím jeho element nebo atribut, načítám si operátor podmínky a literál s kterým porovnávám. Všechno to dělám regulárními výrazy a ukládám si to do asociativního pole `$queryPart`.

Když je jako operátor podmínky části `SELECT` zadané klíčové slovo `CONTAINS`, tak pomocí regulárního výrazu kontroluji, zda je literál řetězcem. Využívám u toho to, že řetězec je v uvozovkách.

Vyhodnocení dotazu

Dotaz vyhodnotím pomocí funkce `xpath()` z třídy `SimpleXMLElement`. Nejprve si ze zadaných dat vytvořím dotaz pro funkci `xpath()`. Pak vytvořený dotaz zadám do funkce `xpath()`, která mi vrátí pole objektů `SimpleXMLElement`, na základě dotazu.

Jako první vyhodnotím část `FROM` a uložím si pouze první objekt v poli. Když je zadaná část `WHERE`, tak ji vyhodnotím jako druhou a nakonec vyhodnotím část `SELECT`.

Závěr

Projekt jsem vyvíjel na referenčním školním serveru merlin, takže jsem měl zaručenou funkčnost mého řešení a nemusel jsem hledat jak nainstalovat starší verzi interpretu PHP na svůj počítač.

K testování jsem využil zveřejněné ukázkové testy, které jsem upravil na automatické porovnání XML souborů a doplnil jsem je o další testy.