

Úvod

Tato dokumentace přibližuje řešení prvního projektu do předmětu IPP. Jedná se o jednoduchý skript v jazyce PHP ve verzi 5.6. Hlavním účelem tohoto projektu je vyhodnocení dotazů nad vstupním XML souborem. Vykonávané dotazy jsou velice podobné příkazu SELECT jazyka SQL.

Tato dokumentace popisuje způsob řešení. Zpracování jednotlivých parametrů, ověření a následně aplikace dotazu nad vstupními daty.

Zpracování parametrů

K zpracování parametrů jsem využil standardní funkci `preg_match()`. Tato funkce dokáže zpracovat parametry podle zadaných regulérních výrazů. Následně se tyto parametry uloží do asociativního pole. V něm je klíčem každého parametru samotný název parametru, a hodnotou prvku pole je konkrétní hodnota parametru.

Při kontrole se dále kontroluje, jestli některý z parametrů, nebyl zadán více krát. To kontroluji tak, že počet prvků pole s parametry se musí shodovat s počtem prvků pole `argv-1` (-1 protože aj název skriptu je v `argv`). Kontroluje si i výskyt parametru `--help`, který musí být zadán samotně. Parametre `--qf` a `--query` se nesmí kombinovat. Tyto skutečnosti kontroluji pomocí funkcí `count()`, `isset()` a `array_key_exists()`. Dále kontroluji, jestli byl zadán alespoň jeden z parametru sloužící k zadání dotazu (`--qf`, `--query`).

Při zpracování parametrů byl největší problém možnost krátkých i dlouhých parametrů. Proto jsem v mém skriptu nepoužil klasickou funkci PHP, a to `getopt()`. Namísto toho jsem všechny parametry zpracoval ručně.

Načtení vstupního XML souboru

Vstupní data se nacházejí v souboru zadaným v parametru `--input`. Danou cestu provedeme pomocí funkce `realpath()` na absolutní. Posléze zjistíme pomocí funkce `is_link()`, jestli je cesta zadána symbolickým odkazem. Když je, využijeme funkci `readlink()`, ta zjistí skutečnou cestu k souboru. Tento postup je použit ne jen při načítání vstupního souboru, ale i při načítání dotazu z parametru `--qf`, nebo hledání výstupního souboru. Jestli není zadán parametr `--input`, očekává se vložení vstupních dat ze standardního vstupu `stdin`.

Po zpracování cesty k souboru použijeme funkce `fopen()` a `fread()` pro načtení vstupu ze souboru do proměnné. Následně z proměnné vytvoříme nový objekt typu `SimpleXMLElement`. Při tom zkontrolujeme správnosti formátu XML. Díky tomuto objektu se poté dá jednoduše pracovat s daty v XML formátu.

Ověření dotazu

Dotaz se skládá z částí `SELECT`, `FROM`, `WHERE` a `LIMIT`. Části `WHERE` a `LIMIT` jsou volitelnou součástí. Na ověření správnosti zadaného dotazu použijeme hlavně práci s regulárními výrazy. Po rozkouskování jednotlivých částí dotazu vložím potřebné informace do asociativního pole `$parsed`. Každý dotaz musí splňovat podmínky vstupní gramatiky ze zadání.

Když nastane situace, kde za částí `SELECT` následuje `CONTAINS`, tak pomocí regulárního výrazu zjistíme, jestli je literál řetězcem. Využijeme u toho to, že řetězec musí být v uvozovkách.

Aplikování dotazu

Ze zadání vychází, že nesmíme použít standardní funkci nato uzpůsobenou `xpath()`. Proto jsme byli nuceni vytvořit si vlastní repliku této funkce. Funkce se nazývá `my_xpath()`. Pracuje na principu rekurzivního procházení objektů `SimpleXMLElement`. Funkce na základě hledaného elementu vrátí pole objektů `SimpleXMLElement`. Tím pádem je splněno zadání a problém nemusíme řešit zbytečně komplikovaně.

Jako první se vyhodnotí část `FROM`, vstupní data se tak „obřežou“ a uloží. Následně se vyhodnotí část `SELECT`, a nakonec i část `WHERE`, když je to požadované.

Výpis výstupních dat

Nakonec se vyhodnotí parametry `--root` a `-n`. Data se upraví podle potřeb těchto parametrů. Následně se výstupní data zapíšou do souboru z parametru `--output`. Když tento parametr není použit, výstup se vypíše na standardní výstup `stdout`.

Závěr

Při pracování na projektu jsem využíval školní server `merlin`. Tímto jsem si ulehčil práci a nemusel jsem hledat způsob, jak nainstalovat starší verzi interpretu PHP na můj počítač. K práci jsem využíval i dokumentaci k PHP jazyku dostupnou z internetových stránek www.php.net.

Pro testování projektu jsem využil ukázkové testy ze stránek předmětu. Testy jsem si upravil podle mého gusta a doplnil o porovnání XML souborů, ale i o další testy. Dále jsem použil testy zveřejněné kolegy, které jsem kvůli změně v zadání trochu poměnil.