Dokumentace úlohy XQR: XML Query v PHP 5 do IPP 2014/2015
Jméno a příjmení: Roman Halík
Login: xhalik01

# Úvod

Tato dokumentace se věnuje principu fungování skriptu, napsaném v jazyce PHP, jehož účelem je filtrace vstupního souboru dle zadaného SQL dotazu. Zároveň skript umožňuje (ne)generování XML hlavičky nebo obalení výsledku filtrace do kořenového elementu, jehož název je určen uživatelem podle vstupního parametru. Dotaz může být zadán jako součást parametru, případně v souboru. Vstupní XML dokument může být přebírán ze standardního vstupu, nebo ze souboru a výstup je také možno uložit do souboru, nebo vypsat na standardní výstup. Nejsou implementována žádná rozšíření.

Skript lze rozdělit na několik částí. Zpracování vstupních parametrů, zpracování dotazu a filtrování vstupního XML dokumentu dle zadaných kritérií. Skript nejprve zkontroluje správnost parametrů, a pokud je vše v pořádku, předává řízení funkci pro zpracování SQL dotazu. Následně se spustí funkce pro filtraci vstupního souboru, vypíše se výsledek a skript se ukončí.

Jestliže v jakémkoli místě skriptu dojde k chybě, je volána funkce `printError`, která klasifikuje typ chyby, vypíše chybové hlášení na standardní chybový výstup a ukončí provádění skriptu.

## Zpracování vstupních parametrů

Parametry jsou zpracovávány ručně. Nejprve se rozdělí do pole a následně se v cyklu kontroluje jeden parametr po druhém pomocí konstrukce `switch`, kdy je k poli pomocí funkce přistupováno jako k zásobníku. S každým průchodem se tedy odebere jeden prvek zásobníku, a zjišťuje se, o jaký parametr se jedná. Pokud není nalezena shoda, generuje se chyba. U parametrů s hodnotou se tato hodnota uloží pro pozdější použití. Stejně tak se uchovává informace o tom, zdali byl parametr zadán, jelikož se žádný parametr nesmí opakovat. Pořadí parametrů není pevně dané. Zároveň je třeba kontrolovat kolize parametrů, které nesmí být zadané současně. U parametrů vyžadující otevření souboru se tato operace provede.

## Analýza a zpracování SQL dotazu

### Funkce parseQuery

SQL dotaz se taktéž kontroluje ručně. Dotaz se rozdělí do pole a každá jeho část se kontroluje zvlášť za využití funkce umožňující používání pole jako frontu. V cyklu se prochází jednotlivé položky a kontroluje se syntaxe i sémantika a zároveň se ukládají hodnoty, nutné pro pozdější filtraci funkcí `xmlFilter`. Nakonec se provede kontrola pořadí klauzulí a hodnoty jednotlivých klauzulí se vrátí jako pole.

### Funkce xmlFilter

Tato funkce přebírá na vstupu pole hodnot z funkce `parseQuery`, které zpracovává. Jejím úkolem je podle těchto hodnot určit podobu výrazu pro funkci `xpath`, který zajišťuje samotnou filtraci vstupního XML souboru. Dále tato funkce realizuje výpis hlavičky, obalení celého výsledného dokumentu kořenovým elementem, a celý výsledek dále posílá na standardní výstup, nebo jej zapíše do souboru. Nakonec funkce zavře otevřené soubory a skript se ukončí s návratovou hodnotou 0.

Problematickým bodem je zpracování jednotlivých klauzulí, které mohou mít různé podoby. Například klauzule `FROM` může nabývat podob „`element`“, „`element.attribute`“ a „`.attribute`“, přičemž pro každý případ je třeba zajistit zcela odlišné chování. Tento problém lze vyřešit samostatným generováním části výrazu pro každou jednu podobu klauzule `FROM`. Stejně tak je třeba zajistit správné fungování nepovinných klauzulí, například `LIMIT`, která zkrátí výsledný výčet elementů na zadanou hodnotu. Tento problém je řešen prostým zastavením výpisu výsledku dotazu po daném počtu elementů.