

Dokumentace úlohy JSN : JSN2XML v php 5.3.3 do předmětu IPP 2015/2016
Jméno a příjmení: Tibor Dudlák
Login: xdudla00

Dokumentace k projektu IPP 2015/2016, JSN: JSON2XML

Analýza zadání

Cílem tohoto projektu bylo vytvořit skript v jazyce PHP (5.3.3), který bude generovat výstupní soubor ve formátu XML podle vstupu ve formátu JSON. Podobu výsledného souboru je možné upravovat spuštěním skriptu se předem definovanými přepínači. Je možné vygenerovat i soubor neodpovídající přesné specifikaci XML.

Postup řešení

Po spuštění skriptu jako první provádím kontrolu zadaných přepínačů. Následně skript udělá kontrolu vstupního a výstupního souboru. Pokud skript neskončí s chybou pokračuje funkcemi které sou rekurzivně volány.

Zpracování parametrů

K zpracování parametrů používám vlastní funkce. Inspiroval sem se funkcí `getopts()`, která vrací asociativní pole. Výsledek mé funkce `parse_args()` jsem následně jako vstupní parametr prodal funkci `check_args()`, která zjišťovala, zda byli nastaveny přepínače obsahující hodnoty potřebné k běhu skriptu. Při zadání přepínačů nasavujících jména elementů ich bylo potřebné skontrolovat funkci `check_name()`, která mohla skončit definovanou chybou (50) a následně znaky nahradit pomocí `preg_replace()` řetězcem, což mohlo vézt k definovaným chybám (51). Pokud nebyli parametry nastaveny, přidala funkce výchozí hodnoty do téhož asociativního pole včetně standardního vstupu/výstupu místo vstupního/výstupního souboru. Výsledkem těchto funkcí je asociativní pole obsahující veškeré potřebné hodnoty, nebo `false` při chybě.

Zpracovávání vstupního souboru

Nejdříve se provede kontrola souborů pro čtení a zápis zda jsou čitelné a v případě JSON i validní. Je-li vstupní soubor validní JSON, což ověří funkce `json_decode()`, je práce se vstupem zcela u konce, hlavně díky tomu, že návratová hodnota funkce je asociativní pole obsahující celou strukturu vstupního JSON souboru.

Generování výstupního souboru

Generování výstupu jsem řešil rekurzivním voláním třech funkcí `write_value()`, `write_object()` a `write_array()` na základě toho, jaký je typ vstupní proměnné. Objekt `XMLWriter` a jeho metody sem v nich dále používal na základě zadání. Důležité bylo správně Začínat a ukončovat tagy co se zanořením a vynořením z rekurze řešilo jednoducho.

Závěr

Projekt jsem vyvíjel na vlastním pc s využitím kompatibilní verze PHP pro systém Ubuntu, využitím repozitáře na [github](#) a také využití skriptu `test.php`, který sloužil k porovnávání výstupu skriptu se soubory, které jsem upravil dle zadání a následně jsem přidal referenční testy. Testování skriptem se provádělo automaticky, za pomoci `circleCI` při nahrání nové verze skriptu na servery github-u spolu se skriptem `is_it_ok.sh`. Testy a skript jsem průběžně zpouštěl na serveru merlin.