

# Logika kódu validátoru a databáze

## Validátor

Podobně jako u JS, byla vytvořena třída Validator pro usnadnění validace formulářových polí. Obsahuje pole \$errors, které si ukládá zaznamenané chyby pro jednotlivé klíče formulářových polí.

Pro získání hodnoty pole při POST dotazu zavolá view soubor funkci validátoru getFromPOST, který vytvoří prázdné pole pro daný klíč, zjistí, zda je nějaká hodnota uložena v \$\_POST pod tímto klíčem a vrátí ji, v opačném případě vrátí prázdný string, což je důležité pro logiku znovu vyplňování formulářů, kde při prvním zobrazení není žádná hodnota k zobrazení dostupná.

Dále obsahuje funkce pro různé typy validací, aby se vyhnulo opakování logiky validace na různých stránkách (register.php, password\_reset.php):

- checkLength – délka poslaného řetězce
- checkEmpty – zda je pole prázdné
- checkContainsNumber – zda obsahuje číslo
- ...

Dále implementuje funkci success(), která vrátí true/false hodnotu, zda byla nalezena chyba v nějakém zaznamenaném poli.

## Databáze

### Třídy objektů databáze

Třída databáze implementuje logiku potřebnou pro práci s databází, ale taktéž lze v tomto souboru nalézt tři třídy reprezentující jednotlivé objekty (article, user, comment), které jsou návratovou hodnotou funkcí, které jsou určeny pro view soubory. Umožňují tak souborům se stránkami jednoduchý přístup k datům těchto objektů bez potřeby znát jejich strukturu v databázi.

Dále třída User obsahuje funkce pro kontrolu role uživatele isAdmin() a isWriter().

### Třída databáze

Obsahuje několik privátních proměnných a funkcí:

- \$file\_folder\_path = složka, kde jsou uloženy jsonové soubory databáze a obrázky
- funkce pro vytvoření objektů databáze ze struktury pole v json souborech
- funkce pro načtení a zápis dat z/do příslušného souboru

Mezi veřejné funkce patří funkce určené pro view soubory. Různé funkce typu add, remove, edit a get. Také funkce saveImage, která uloží daný soubor z superglobální proměnné \$\_FILES do databáze ve třech různých velikostech (vlastnost Article \$image\_path je pak pouze relativní cesta k obrázku a stránky načítají soubor s tímto jménem ve složce s požadovanou velikostí).

Co se týče jednotlivých funkcí, tak jejich funkcionalita je popsána ve zdrojové dokumentaci. Důležitá je ale struktura databáze. Všechny objekty (article, user, comment) jsou uloženy v samostatných json souborech ve formátu klíč-pole dat. V případě uživatelů je klíč jejich uživatelské jméno, které je unikátní, což by mělo být zaručeno v kontrole registračního formuláře. Dále articles a comments mají vlastní unikátní id vygenerováno přes funkci uniqid().

Tyto objekty jsou propojeny mezi sebou pomocí jejich identifikátorů. Například komentáře napsané uživatelem lze dohledat pomocí id komentářů uložených v seznamu v poli dat uživatele v users. Stejně tak komentáře pod daným článkem lze také dohledat pomocí id uložených v poli article pod daným articleId. Komentář zase naopak má uloženo, kdo ho napsal a pod který článek patří.