Szczegółowa koncepcja proponowanego rozwiązania i dokumentacja techniczna

Projekt – RSO MongoDB

Autorzy:

* Tomasz Adamiec
* Piotr Cebulski
* Marek Kowalski
* Mateusz Rosiewicz
* Paweł Sokołowski
* Marcin Wnuk

Warszawa, 2013

CRUD

* 1. FileOperations - Operacja na plikach

Klasa FileOperations odpowiada za czynności wykonywane na plikach. Jest to klasa zawierająca najczęściej powtarzane czynności, zapewniająca tworzenie i modyfikację struktury danych. Struktura bazy danych tworzona jest w folderze domowym użytkownika w folderze „exampleDB”

Zaimplementowane funkcje:

* Wczytywanie bajtów z pliku reprezentujących BSON dokument.
* Wyszukiwanie pliku zawierającego daną wartość ObjectID.
* Wyszukiwanie pliku po nazwie pliku i nazwie kolekcji.
* Tworzenie pliku na dysku.
* Zwrócenie listy plików z danego folderu.
  1. Selector -Wybór plików z zapytania

Klasa Selector posiada funkcje zwracające listę ID plików wymienionych w zapytaniu. Porównywane są dokumenty zapisane w plikach jak i dokument stanowiący pole selektora.

* 1. Insert - Zapisywanie plików na dysku

Funkcja odpowiedzialna za dodanie danych wprowadzonych przez użytkownika do bazy przyjmuje, jako argument listę dokumentów stanowiących pole obiektu InsertMessage. Dla każdego elementu zostaje wywołana metoda zapisująca dokument na dysk w postaci bajtów. W zależności czy dana kolekcja wymieniona w InsertMessage istnieje, tworzona jest nowa, bądź aktualizowana istniejąca. Pozostałe operacje opierają się o wyżej opisaną klasę FileOperations.

* 1. Delete - Usuwanie plików z dysku

Delete dzieli się na dwa etapy. Pobranie przez Selector listy plików wybranych przez użytkownika do usunięcia oraz wyszukanie wybranych plików i ich usunięcie.

* 1. Update - Modyfikacja plików na dysku

Realizacja update polega na pobraniu nazwy pliku do zmodyfikowania z updateMessage i wyszukaniu go na dysku. Następnie dane są porównywane. Modyfikacja to stworzenie nowego dokumentu zawierającego niekonfliktowe pola oraz dodanie modyfikacji z updateMessage. Na koniec kasowany jest stary plik i tworzony nowy zawierający zmodyfikowany dokument.

BSON

* 1. Czym jest BSON

BSON (skrót od Binary JSON) jest binarnym formatem stosowanym do przechowywania dokumentów. Każdy dokument składa się z zestawu elementów typu nazwa-wartość. Gdzie wartość może należeć do wielu różnych typów (takich jak int, double, string) w tym typów złożonych takich jak dokument w dokumencie czy tablica elementów.

* 1. Zastosowanie BSON

W naszym projekcie BSON stosowany jest do przesyłania, zapisywania i przeszukiwania zawartości bazy danych. Pole "\_id", które każdy BSON zawiera, stosowane jest, jako klucz główny ze względu na swoją unikalność.

Opis rozwiązania

* 1. Ogólny zarys

Rozwiązanie modułu BSON składa się z pięciu klas:

* BSON - klasa abstrakcyjna zawierająca statyczne metody służące do parsowania tablicy bytów zawierającej BSON na BSONDocument
* BSONDocument - sparsowany dokument BSON, zawierający obiekty BSONElement różnych typów
* BSONElement - pojedyncza para nazwa - wartość
* BSONtype - wyliczenie możliwych typów wartości zawartych w BSON
* ObjectID - pole ObjectID dokumentu BSON, zawiera 4 numeryczne pola: czas utworzenia, id maszyny, id procesu, licznik

Najważniejszymi elementem rozwiązania jest klasa BSON. Zawarte w niej metody parseBSON i getBSON stanowią podstawę działania całego pakietu.

* 1. Parsowanie BSON

Parsowanie BSON (czyli przejście z tablicy bajtów na obiekt klasy BSONDocument) wykonywane jest przez metodę parseBSON. Najpierw odczytywane jest pierwsze pole dokumentu BSON, czyli jego długość.

Następnie dla każdego elementu (pary nazwa-wartość) zawartego w dokumencie proces wygląda tak samo. W pierwszej kolejności odczytywany jest jego typ i nazwa. Parsowanie wartości przechowywanej w elemencie jest zależne od typu i jest wykonywane przez metodę parse klasy BSON.

* 1. Tworzenie BSON

Tworzenie BSON (czyli tablicy bajtów z obiektu klasy BSONDocument) jest procesem odwrotnym do jego parsowania. W pierwszej kolejności tworzona jest lista, w której będą przechowywane tablice bajtów odpowiadające poszczególnym elementom zawartym w dokumencie.

Kiedy wszystkie elementy są już przetworzone na tablice bajtów (zajmuje się tym metoda getByte) obliczana jest łączna długość wszystkich tablic. W ostatnim kroku wszystkie tablice łączone są w jedną całość, do której dodawana jest na początku łączna długość i na końcu bajt zerowy.

[CRUD 2](#_Toc358638587)

[1.1 FileOperations - Operacja na plikach 2](#_Toc358638588)

[1.2 Selector -Wybór plików z zapytania 2](#_Toc358638589)

[1.3 Insert - Zapisywanie plików na dysku 2](#_Toc358638590)

[1.4 Delete - Usuwanie plików z dysku 2](#_Toc358638591)

[2.1 Update - Modyfikacja plików na dysku 2](#_Toc358638592)

[BSON 3](#_Toc358638593)

[2.1 Czym jest BSON 3](#_Toc358638594)

[2.2 Zastosowanie BSON 3](#_Toc358638595)

[Opis rozwiązania 3](#_Toc358638596)

[3.1 Ogólny zarys 3](#_Toc358638597)

[3.2 Parsowanie BSON 3](#_Toc358638598)

[3.3 Tworzenie BSON 4](#_Toc358638599)