

Opis testów do ćwiczenia 3 branch Rest Client w klasie ProductServiceTest.java

1. `retrieveAllProducts_shouldReturnEmptyListWhenNoProductsFound` – test zwraca nam pustą listę kiedy nie znajdzie żadnych produktów. Tworzymy listę produktów i dodajemy do niej wszystkie produkty z `productService`. `assertEquals` porównuje będące parametrami obiekty i rzuca wyjątek, jeżeli obiekty te nie są sobie równe. W tym przypadku 0 z rozmiarem listy który powinien wynosić 0 jeśli test ma się powieść.
2. `retrieveAllProducts_shouldReturnProductListWhenProductsFound` – powinien nam zwrócić listę jeśli znajdzie produkty. W tym przypadku powinna ona mieć rozmiar 2. Porównujemy `assertEquals`. Tworzymy listę produktów a następnie dodajemy je do `serverApp`. Tworzymy kolejną listę i przypisujemy do niej wszystkie produkty. Powinna mieć 2.
3. `retrieveProducts_shouldReturnEmptyListWhenNoProductsFound` – zwraca pustą jeśli nie znajdzie jakiegoś produktu który będzie pasował dla szukanego typu. Stworzona lista ma przypisane produkty
4. `retrieveProducts_shouldReturnProductListWhenProductsFound` – powinien zakończyć się sukcesem kiedy dostaniemy listę która będzie zawierać 4 elementy z produktami o podanych parametrach.
5. `storeNewProduct_shouldCreateNewProduct`- Niestety nie do końca rozumiem w tej metodzie się dzieje. Tworzę listę produktów. Tworzę listę produktów następnie listę id i porównuje każdy element tablicy z stworzonymi id. (niestety tego nie rozumiem.)
6. `storeNewProduct_shouldRaiseAnExceptionIfIDFieldSet` – metoda powinna zwracać wyjątek jeśli tworzony produkt ma podany przy tworzeniu id.
7. `updateProduct_shouldUpdateProductIfFound` – metoda powinna zrobić update produktu jeśli go znajdzie. Tworzymy produkt. Nadajemy mu własności, następnie tworzymy kolejny obiekt który będzie miał takie samo ID i robimy update produktu. Porównujemy potem 2 obiekty.
8. `updateProduct_shouldThrowExceptionIfProductNotFound`- metoda powinna rzucić wyjątkiem jeśli nie znajdzie produktu który może updatować. Tworzymy jeden produkt i chcemy zrobić update na nim ale on nie może się updatować bo nie został dodany do `productService`.
9. `deleteProduct_shouldDeleteProduct` – metoda która powinna usunąć produkt. Tworzymy a następnie usuwamy produkt a na końcu porównujemy czy rozmiar jest równy 0.
10. `deleteProduct_shouldThrowExceptionIfProductNotFound` – metoda ta powinna rzucić wyjątkiem przy nie znalezieniu produktu który ma zostać usunięty. Produkt jest tworzony ale nie jest dodwany do `ServerApplication` a następnie próbujemy go usunąć.
11. `retrieveProduct_shouldRetrieveProductIfExists` – Zwracanie produktu jeśli istnieje. Tworzymy obiekt produkt następnie zmienną `int` i przypisujemy do niej `serverApp.addProduct(product)`. Kolejny obiekt `result` ma być takim samym produktem który stworzyliśmy wcześniej.
12. `retrieveProduct_shouldThrowExceptionIfProductNotFound` – metoda powinna rzucić wyjątkiem kiedy nie znajdzie produktu. Zwracamy produkt o indeksie 1 ale wcześniej żadnego nie tworzymy i nie dodajemy.