

Page Object Model

Maciej Skwara, Adrian Rymut


16.06.2019



Wzorzec projektowy Page Object jest obecnie jednym z najpopularniejszych podejść do automatyzacji testów webowych. Znajduje zastosowanie we frameworkach tworzonych według paradygmatu programowania obiektowego i pozwala na pełne wykorzystanie jego zalet. Na czym polega Page Object i w czym dokładnie może pomóc?

Z punktu widzenia deweloperów oprogramowania testowanie może wydawać się zajęciem nieco nudnym i monotonnym. Powszechne wyobrażenia na temat tego, czym zajmują się testerzy, często powielają obraz testera-klikacza. Testy automatyczne, w opozycji do testów manualnych, w dużej mierze opierają się na programowaniu. Zadaniem testera jest przygotowanie narzędzi, które w sposób jak najbardziej autonomiczny wykonają zadany scenariusz i ocenią jego rezultaty.

BENEFITS of Automation Testing



Ważną zaletą automatycznego wykonywania testów jest ich powtarzalność. W przypadku scenariusza wykonywanego przez człowieka wręcz niemożliwe jest wymuszenie powtarzania jego kroków dokładnie w ten sam sposób za każdym razem. Wraz z upływem czasu organizm człowieka się męczy, spada koncentracja i coraz trudniej jest mu ustrzec się błędów. Natomiast poprawnie przygotowany test automatyczny gwarantuje niemal całkowitą powtarzalność wykonywanych akcji.

Powtarzalność wykonywania wiąże się także z systematycznością. Testy automatyczne mogą być uruchamiane cyklicznie (np. codziennie wraz z daily buildem oprogramowania). Przede wszystkim testy automatyczne świetnie uzupełniają testy manualne. Trzeba jednak podkreślić słowo uzupełniają. Mylne jest mniemanie, że testy automatyczne mogą całkowicie zastąpić i wyeliminować z użycia testy manualne.

WebDriver oferuje dwie kluczowe funkcjonalności, które czynią to narzędzie wartym uwagi. Z jednej strony to bardzo wygodne dla programisty API pozwalające na interakcję z przeglądarką, z drugiej zaś to koncepcja sterowników (ang. driver), które tę bezpośrednią komunikację umożliwiają.



Selenium WebDriver

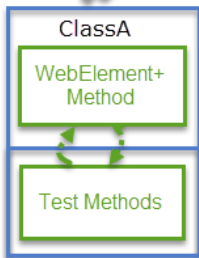
WebDriver oferuje obsługę następujących przeglądarek:

- Internet Explorer,
- Firefox,
- Chrome

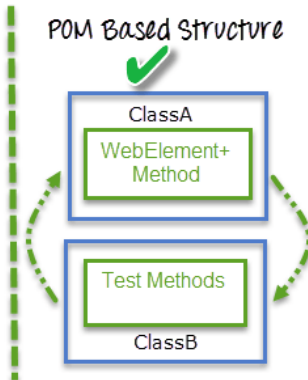
Selenium WebDriver oferuje głównie szereg klas i metod pomagających nam porozumieć się z przeglądarką.

Najważniejszą ideą, stojącą u podstaw wzorca Page Object, jest opakowanie strony i jej elementów w odpowiednie klasy i ich pola. Należy pamiętać, że wbrew temu, co sugeruje powszechnie przyjęte nazewnictwo, klasą niekoniecznie musi być cała strona – w praktyce częściej zdarza się, że jest nią określona funkcjonalna część strony, na przykład panel czy lista wyników wyszukiwania.

Non POM Structure



POM Based Structure



W typowym przypadku kod rozdzielony jest na klasy opakowujące testowane strony (przykładowo zgromadzone w folderze o nazwie *Pages*) i na scenariusze testowe (znajdujące się na przykład w folderze *Tests*). Testy korzystają z obiektów klas z folderu *Pages*, aby wykonywać poszczególne kroki na stronie.

