

Hubert Makowski WCY21KY1S1

Git - sposób „comitowania” zmian

Jeżeli chcemy skorzystać z „commitowania” najpierw musimy przeprowadzić inicjalizację repozytorium w ścieżce (katalogu) musimy stworzyć nowy podkatalogu .git który zawiera wszystkie pliki repozytorium (szkielet) poprzez użycie komendy git init.

W celu sprawdzenia które pliki są w jakim stanie (staged, not staged) używa się komendy git status.

Przed przeprowadzenie „commita” plik repozytorium musi znajdować się w stanie śledzonym. Domyślnie pliki są nieśledzone, więc aby śledzić plik należy skorzystać z komendy git add. Po przeprowadzeniu „commita” (komenda git commit) pliki które zostały zmieniane od ostatniego „commit” są traktowane jako zmodyfikowane. Jeśli chcemy poinformować system kontroli, że ostatni zmiany w pliku od ostatniego „commit” mają zostać uwzględnione w następnym „commit”, należy użyć komendy git add. Kiedy zmieniamy śledzony plik i uruchamiamy komendę git status, otrzymamy powiadomienie, mówiące, że śledzony plik został zmieniony. Trzeba pamiętać że jeśli chcemy uruchomić komendę git commit, to pliki które nie zostały zapisane w konkretnej fazie po ich modyfikacji (zmodyfikowaliśmy plik i nie użyliśmy komendy git add) nie zostaną „commitowane”. Przed „commitowaniem” należy sprawdzić na którym rozgałęzieniu (branch) pracujemy, ponieważ „commit” na jednym rozgałęzieniu nie oznacza „commit” na drugim rozgałęzieniu. Jeśli chcemy zapisać stan każdego pliku w repozytorium przed „commit” można użyć komendy git commit -a. Kiedy chcemy usunąć plik z systemu kontroli GIT należy użyć komendy git rm. Ciekawym rozwiązaniem w śledzeniu „commit” jest komenda git log umożliwiająca sprawdzenie historii „commit-ów” (śledzimy dane takie jak data, autor „commita”). Kiedy popełnimy błąd w „commitowaniu” (zapomnimy dodać pliku) można użyć komendy git commit --amend w celu przywrócenia stanu sprzed ostatniego „commit”. Jeżeli chcemy zmienić ostatni zapisany stan pliku (kiedy użyliśmy komendy git add) można użyć komendy git reset HEAD.